OPTIMIZATION OF A TRAFFIC SIGNAL SYSTEM THROUGH COMPUTER SIMULATION

JUNE 1972 — NUMBER 8

BY

GARY W. DAVIES

JHRP

JOINT HIGHWAY RESEARCH PROJECT
PURDUE UNIVERSITY AND
INDIANA STATE HIGHWAY COMMISSION

Final Report

OPTIMIZATION OF A TRAFFIC SIGNAL SYSTEM THROUGH COMPUTER SIMULATION

TO: J. F. McLaughlin, Director
Joint Highway Research Project

FROM: H. L. Michael, Associate Director
Joint Highway Research Project

June 22, 1972

Project: C-36-17JJ

File: 8-4-36

The attached Final Report titled "Optimization of a Traffic Signal System Through Computer Simulation" is presented for acceptance as fulfillment of the objectives of the research proposed and approved as a JHRP study. Mr. Gary W. Davies, Graduate Assistant in Research on our staff, authored the Report under the direction of Professors W. L. Grecco and K. W. Heathington.

The Report introduces a simulation model, SIGNET, which was developed. The model is useful in signal timing studies and in evaluating many other traffic engineering measures, such as turn prohibitions, parking restrictions, etc. The research demonstrated that use of the model could make a signal system timing design more efficient and obtained more rapidly.

The Report is presented for acceptance.

Respectfully submitted,

Harold L. Michael

Harold L. Michael
Associate Director

HLM:ms

cc: W. L. Dolch
R. L. Eskew
W. H. Goetz
W. L. Grecco
M. J. Gutzwiller
G. K. Hallock
M. E. Harr
R. H. Harrell
M. L. Hayes
R. D. Miles
J. W.Miller
C. F. Scholer
M. B. Scott
J. A. Spooner
N. W. Steinkamp
H. R. J. Walsh
K. B. WOods
E. J. Yoder

Final Report

OPTIMIZATION OF A TRAFFIC SIGNAL SYSTEM THROUGH COMPUTER SIMULATION

by

Gary W. Davies
Graduate Assistant in Research

Joint Highway Research Project

Project No.: C-36-17JJ

File No.: 8-4-36

Conducted By

Joint Highway Research Project
Engineering Experiment Station
Purdue University

In Cooperation with

Indiana State Highway Commission

Purdue University
Lafayette, Indiana
June 22, 1972

Digitized by the Internet Archive
in 2011 with funding from
LYRASIS members and Sloan Foundation; Indiana Department of Transportation

http://www.archive.org/details/optimizationoftr00davi

ACKNOWLEDGMENTS

It would be a prodigious task to name and thank everyone who contributed to this research effort. My special thanks must go to a few, however, for without their guidance and assistance this work would have been impossible.

I would like to thank the Joint Highway Research Project for giving me the opportunity to do this research; Dr. William L. Grecco, for his guidance in choosing and developing the topic and performing the research; Dr. Kenneth W. Heathington, for his advice and encouragement throughout the project; Dr. I. W. Burr for his guidance in the statistical aspects of the project and his review of the manuscript; Professor H. L. Michael for his advice in traffic engineering; George Stafford for his invaluable assistance in collecting the needed traffic data; and Marian Sipes for her many favors while preparing the manuscript.

To all of these people go my most sincere thanks.

TABLE OF CONTENTS

Page

LIST OF TABLES.................................. v

LIST OF FIGURES................................. viii

ABSTRACT.. xi

CHAPTER I. INTRODUCTION............................ 1

CHAPTER II. DEVELOPMENT OF THE SIMULATION MODEL... 6

- Mode of Representation......................... 10
- Mathematics of Vehicle Behavior................ 15

CHAPTER III. WORKING DETAILS OF THE SIMULATION MODEL.. 49

- Geometric Configuration........................ 53
- SIGNET Program Structure....................... 66
- Simulation Output.............................. 93

CHAPTER IV. THE STREET NETWORK..................... 96

CHAPTER V. MODEL TESTING AND VALIDATION........... 101

CHAPTER VI. SIGNAL TIMING STUDIES.................. 128

CHAPTER VII. CONCLUSION............................ 139

BIBLIOGRAPHY...................................... 141

APPENDICES

- Appendix A - PRESIG Program Details........... 145
- Appendix B - SIGNET Program Details........... 168
- Appendix C - Traffic Flow Data................ 236

LIST OF TABLES

Table Page

1 Amber Decision Probability.................. 28

2 Queue Discharge -- Seconds to Cross Stop Line.................................. 34

3 Lane Change Decision Probability........... 39

4 Left-Turn Decision Probability............. 45

5 Signal Indications......................... 51

6 Analysis of Variance, Effect of Volume on Average Delay.............................. 110

7 Analysis of Variance, Effect of Volume on Average Travel Time........................ 111

8 Analysis of Variance, Effect of Volume on Average Speed.............................. 112

9 Analysis of Variance, Effect of Volume on Average Queue Length (Link 57)............. 113

10 Analysis of Variance, Effect of Target Velocity on Average Travel Time............ 114

11 Analysis of Variance, Effect of Target Velocity on Average Speed.................. 115

12 Analysis of Variance, Effect of Effective Length on Average Travel Time............. 116

13 Analysis of Variance, Effect of Effective Length on Average Speed................... 117

14 Analysis of Variance, Effect of Trucks on Average Travel Time....................... 118

15 Analysis of Variance, Effect of Trucks on Average Speed............................. 119

LIST OF TABLES (Continued)

Table page

16 Analysis of Variance, Effect of Maximum Desired Acceleration of Average Travel Time.......................... 120

17 Analysis of Variance, Effect of Maximum Desired Acceleration on Average Speed...... 121

18 Analysis of Variance, Effect of Characteristic Speed on Average Speed...... 122

19 Analysis of Variance, Effect of Characteristics Speed on Average Speed..... 123

20 Analysis of Variance, Effect of Proportionality Coefficient on Average Travel Time................................ 124

21 Analysis of Variance, Effect of Proportionality Coefficient on Average Speed......... 125

22 Average Delay Caused by Signal Timing Alternatives, sec./veh...................... 133

Appendix
Table

A-1 Default Radii Used In PRESIG............... 150

A-2 PRESIG Punched Card Input Format........... 151

A-3 PRESIG Program Code........................ 155

B-1 SIGNET Punched Card Input Format........... 180

B-2 SIGNET Program Code........................ 188

C-1 Network Traffic Signal Settings............ 236

C-2 Tabulation of Traffic Characteristics, Average Conditions, 4:00 to 4:15 P.M....... 242

C-3 Tabulation of Traffic Characteristics, Average Conditions, 4:15 to 4:30 P.M....... 245

C-4 Tabulation of Traffic Characteristics, Average Conditions, 4:30 to 4:45 P.M....... 248

LIST OF TABLES (Continued)

Table Page

C-5 Tabulation of Traffic Characteristics, Average Conditions, 4:45 to 5:00 P.M....... 251

C-6 Tabulation of Traffic Characteristics, Average Conditions, 5:00 to 5:15 P.M....... 254

C-7 Tabulation of Traffic Characteristics, Average Conditions, 5:15 to 5:30 P.M....... 257

LIST OF FIGURES

Figure Page

1 Content of ICAR Word........................ 14

2 Comparison of Negative Exponential and Translated Negative Exponential............ 17

3 General Queue Discharge Relationships...... 25

4 Amber Decision Probability Distribution.... 27

5 Queue Discharge Comparison................. 35

6 Change Zone Definition..................... 37

7 Lane Change Decision Probability Distribution.40

8 Variables Used in Computing Adjusted Stopping Rate.............................. 42

9 Left-Turn Decision Probability Distributions............................. 46

10 A simplified Link Pair..................... 54

11 Compensation for Unequal Discharge Boundaries................................ 54

12 Legal Turning Movements.................... 56

13 Samples of Turning Movement Combinations... 56

14 Contributing Turning Movements............. 58

15 Computation of Discharge Boundary Offset... 59

16 Discharge Boundary Computations............ 63

17 PRESIG Flowchart........................... 65

18 SIGNET Program Structure................... 68

19 Main Program Chain......................... 69

LIST OF FIGURES (Continued)

Figure Page

20 Vehicle Generation Routine (INPLINK)....... 72

21 Vehicle Update Routine (UPDATE)............ 73

22 Lane Change Routine (CHANGE)............... 75

23 Signal Update Routine (SIGNAL)............. 78

24 Amber Decision Routine (AMBER)............. 80

25 General Behavior Routine (GENBHR).......... 81

26 Red Approach Routine (REDAPP).............. 83

27 Right Turn Routine (RIGHT)................. 85

28 Left Turn Routine (LEFT)................... 86

29 Left Turn Conflict Routine (CONLT)......... 89

30 Cross-Street Conflict Routine (CONST)...... 90

31 Lag and Gap Acceptance Routine (GAP)....... 91

32 Street Map of the Test Network............. 97

33 Schematic Diagram of the Test Network...... 98

34 Comparison of Travel Times, Link 28........ 103

35 Comparison of Travel Times, Link 56........ 104

36 Comparison of Travel Times, Link 59........ 105

37 Effect of Volume on Average Delay.......... 110

38 Effect of Volume on Average Travel Time.... 111

39 Effect of Volume on Average Speed.......... 112

40 Effect of Volume on Average Queue Length (Link 57).................................. 113

41 Effect of Target Velocity on Average Travel Time................................. 114

42 Effect of Target Velocity on Average Speed. 115

43 Effect of Effective Length on Average Travel Time................................. 116

LIST OF FIGURES (Continued)

Figure Page

44 Effect of Effective Length on Average Speed.................................... 117

45 Effect of Trucks on Average Travel Time.... 118

46 Effect of Trucks on Average Speed.......... 119

47 Effect of Maximum Desired Acceleration on Average Travel Time......................... 120

48 Effect of Maximum Desired Acceleration on Average Speed............................... 121

49 Effect of Characteristic Speed (a_0) on Average Travel Time......................... 122

50 Effect of Characteristic Speed (a_0) on Average Speed............................... 123

51 Effect of Proportionality Coefficient (K) on Average Travel Time...................... 124

52 Effect of Proportionality Coefficient (K) on Average Speed............................ 125

Appendix Figure

A-1 Intersection Types......................... 146

B-1 Input Data Deck Structure.................. 169

B-2 SIGNET Program Deck Structure.............. 235

ABSTRACT

Davies, Gary William. M.S.C.E., Purdue University, June, 1972. Optimization of a Traffic Signal Network through Computer Simulation. Major Professors: William L. Grecco and Kenneth W. Heathington.

The objective of this study was to develop a simulation program which would be suitable for use in the design of traffic signal systems. In keeping with this, it was desired to develop, program, and test a conceptual model that would simulate the flow of vehicles on a street network controlled by traffic signals. Secondary goals were that the model be general enough to apply to any moderate sized network, and that the final program be such that the inputs and outputs could be easily understood by a non-computer oriented traffic engineer.

A simulation model, SIGNET, was developed to reproduce traffic under laboratory conditions. The program was fairly economical, achieving a 6.5 to 1 real-time to simulation ratio. The model was quite microscopic in nature, since each vehicle was individually processed through the network. A sensitivity analysis indicated that the model simulated traffic flows quite accurately.

The model was used to evaluate several alternative timing plans, and indicated by a significant difference in

mean delay between two of these plans that it is indeed sensitive to variations in signal timing. It was demonstrated that the incorporation of simulation into the design procedure could make that procedure quicker, at the same time resulting in the production of a more efficient timing plan.

The use of the SIGNET model is not restricted to signal timing studies. In fact, the effects of many other traffic engineering measures may be evaluated by varying the program inputs. Such measures could include turning movement prohibitions, parking restrictions, unbalanced lane operation, and one-way street operation. A comprehensive set of improvements to the network can thus be easily evaluated. This capability is important because traffic signal optimization cannot be considered to be the sole means of obtaining maximum network efficiency.

CHAPTER I

INTRODUCTION

Since its first appearance on the urban scene, the automobile has become a dynamic social force molding the structure of cities and the style of daily life. It represents unprecedented individual mobility, comfort, and convenience; consequently it has become enormously popular.

With the increased use of the automobile, however, have come many problems. Congestion and the ensuing delays are just short of intolerable in many cities. A more subtle but perhaps more dangerous problem is the deterioration of the total environment: deaths, injuries, and property damage caused by accidents continue to increase; exhaust fumes and noise provide very real sources of pollution; and automobiles seem to penetrate into every available clear space of land.

Increasing auto traffic and the attendant problems outlined above are threatening the viability of the central business districts of many cities more each year. There are currently in vogue four approaches to the solution of these problems:

1. Road construction and improvement: this is very expensive, tends to generate new demand, and necessitates the provision of additional parking facilities. It involves a net loss of scarce land which might be put to other use.

2. Mass transit construction and improvement: This line of attack is currently being pursued in several major cities. Model preferences seem to be based primarily on personal tastes rather than on personal finances. Therefore the comfort, convenience, and privacy of mass transit vehicles must be improved considerably to at least stabilize transit usage in many urban areas.

3. Auto user constraints: in this area are included special peak-hour tolls, the staggering of office hours, and deliberately increased parking lot charges to discourage auto commuters. Such schemes would certainly tend to reduce critical peak hour volumes, but it is quite possible that the viability of the central business district would also be threatened by such action.

4. Increasing the efficiency of the existing system: before any of the above more drastic measures are implemented there is a general trend to obtain the maximum efficiency out of the existing facilities. Such actions involve, at least in their simpler forms, relatively minor expenditures and little or no construction.

Measures of this kind include turning movement controls at intersections, parking restrictions, unbalanced lane operation, one-way street operation, improvement of signs and markings, and channelization. Another measure, improvement of traffic signal operation, can result in improved use of the intersections, thereby significantly increasing overall network efficiency.

The at-grade intersection is the most critical component of the urban traffic network. It is the focal point of congestion and the greatest source of delay. A traffic signal can allocate the usage of this facility between the conflicting streams of vehicles, and by coordinating the timing of adjacent signals, the flow of traffic can be markedly improved.

To date the theoretical treatment of signal controlled traffic networks has been limited. The determination of optimal signal settings at an isolated intersection was thoroughly covered in a monograph by F. V. Webster (38). The interactive effects of more closely spaced signals on arterials has been considered with varying degrees of success in many methods, including the Yardeni Time-Space Design Model (41), the Little Maximal Bandwidth Model (28), the Delay/Difference-of-Offset Method (15,23,36), the Eagle Signal Method (31), and the Raus Method (31). The problems which these interactive effects raise in closed networks, however,

have slowed progress in that area. One of the more significant efforts to date has been the development of SIGOP, a traffic signal optimization program, by Peat, Marwick, Livingston, and Co. Given a set of data describing the geometrics of a signalized street network and traffic characteristics for a given period of interest, the function of SIGOP is to determine a plan of cycle length, phase splits, and offsets for the traffic signals in the network. Other procedures such as the Webster Method, the Delay/Difference-of-Offset Method, the British Combination Method (36), the Volume Priority Method (36), and the Preferential Street Method (36) may be combined to produce efficient timing plans also. Under different conditions each of these methods could produce the most efficient timing plan; it is necessary to devise a means of choosing this optimal plan, therefore.

Probably the most accurate means of determining the proper timing scheme for a given network would be to develop several alternatives, implement each, and study each scheme's effect on the system. This approach poses several problems, however. It would be very expensive to sufficiently instrument the system to gather the required data; the time required to undertake such a design process would also be prohibitive; it could even be inconvenient to the motorist in extreme cases.

Another method, computer simulation, may be far less expensive than field experimentation, and it offers a much wider scope for innovation and experimentation with the complete control the researcher has over its inputs and operation. It is probable that a design procedure incorporating simulation would also be considerably faster than other procedures.

The objective of this study was to develop a simulation program which would be suitable for use in the design of traffic signal systems. In keeping with this, it was desired to develop, program, and test a conceptual model that would simulate the flow of vehicles on a street network controlled by traffic signals. Secondary goals were that the model be general enough to apply to any moderate sized network, and that the final program be such that the inputs and outputs could be easily understood by a non-computer oriented traffic engineer.

In the following chapters, the development of the simulation model is traced and the basic portions of the model defined (Chapter II), the working details of the model are described (Chapter III), the study network is described (Chapter IV), the testing and validation of the model are explained (Chapter V), and the use of the model in the design of a signal timing plan is shown by example (Chapter VI).

Another method, computer simulation, may be far less expensive than field experimentation and it offers a much wider scope for investigation because of the complete control the researcher has over the inputs and operation. It is probable that a design procedure incorporating simulation would also be considerably faster than other procedures.

The objective of this study was the development of a simulation program which would be suitable for use in the design of traffic signal systems. In keeping with this it was decided to develop, [illegible] a conceptual model that would simulate the action of each vehicle in a street network controlled by traffic signals. Secondary goals were that the model be general enough to allow for any networks sized between small and large and the final program be such that the inputs and outputs could be easily understood by a non-computer oriented traffic engineer.

In the following chapters, the development of the simulation model is discussed and the basic operations of the model defined (Chapter II), the working details of the model are described (Chapter III), the study network is described (Chapter IV), the testing and validation of the model are explained (Chapter V), and the use of the model in the design of a signal timing plan is shown by example (Chapter VI).

CHAPTER II

DEVELOPMENT OF THE SIMULATION MODEL

The use of computer simulation as a traffic research tool is virtually untapped. At a time when businessmen, defense analysts, scientists, corporate managers, and others are employing simulation to assist them in making better decisions and to gain more understanding of complex processes, many traffic engineers have failed to consider seriously the use of simulation.

Many significant efforts, some with varying degrees of success, have been made, however, at closing this breech. Production-type, general purpose computers were not readily available until about 1954. In 1956, three digital computer simulations were reported in the traffic engineering literature. Gerlough discussed the simulation of traffic on a freeway (11). Wong's paper described the simulation of a portion of a multi-lane boulevard (40). Goode, Pollmar, and Wright constructed a model of a signalized intersection (16). Two separate studies of intersection simulation performed by Benhard (4) and Lewis (27) dealt with the intersection of two streets, each with two lanes, and with actuated signal control. The models were greatly simplified in that turning and passing were

prohibited. These and other early simulations permitted only a limited and somewhat arbitrary action of vehicles. Later investigations incorporated several refinements. Research by Kell (25) involved the simulation of the intersection of two urban two-lane streets under various types of traffic control. The much larger problem of simulating a network was first attacked by Stark (33) and Schwartz (32). Besides being slow, the Stark model was very heavily network dependent, meaning it could not be easily applied to any other network. Additionally, the printed output from this model was quite inadequate, since it relied primarily on an on-line visual display device and could not trace individual vehicles nor compute their delays. Although the Schwartz model was quite general, it too relied heavily on deterministic relationships in describing vehicle behavior. Traffic flow theory has advanced to the point where distributions would be more desirable in most of these cases, however.

Probably the most sophisticated model to date is TRANS, developed by Planning Research Corporation (13,14, 15,37). The size and configuration of arterial networks to which TRANS can be applied is very flexible. Either open networks (i.e. arterials) or closed networks can be studied. Wherever available traffic flow data permit distributions are used to describe vehicle actions. The parameters of these distributions are required as input.

In simulating a large, complex network in the District of Columbia consisting of over 200 links, a simulated time to real time ratio of 1:4 was achieved.

A major drawback to the widespread use of TRANS is that it was written in FAP for operation on the IBM 7090 computer. This would tend to make the program highly machine dependent. Several concepts employed in the TRANS model, however, were utilized in developing the model in this study.

There are two basic areas which in the past have slowed the development of simulation as a useful and practical tool for the traffic engineer. The most obvious has been the state of hardware development. Fortunately this problem has been all but resolved with the development of today's high level computers.

A more subtle, but more critical problem is that not enough is known about traffic itself, especially in the area of defining distributions which can be used to predict vehicle actions. Especially critical in this regard are left-turn gap and lag acceptance probabilities, which relate directly to the amount of delay a vehicle encounters while making a left turn. Other distributions needing further definition are the amber acceptance distribution, lane change gap acceptance distributions, and lane distributions at intersections. If the distributions used as a basis for a mathematical model are incorrect, it is

then quite probable that the model will also be incorrect. This problem which has considerably hampered simulation development in the past has been somewhat of a problem in this research. Until much more progress is made in the area of traffic flow theory, it will probably continue to be one.

To be useful, traffic simulation must satisfy three basic considerations:

1. The results of the simulation must fit the facts; in other words, observations obtained as a result of simulation must agree with similar results obtained from observations of actual traffic flow.

2. The time required to simulate a problem must be reasonable; that is, the ratio of simulated time to real time must be such that computer simulation of a street network is economically feasible. Before embarking on a study the objectives of that study should be thoroughly reviewed and the alternative techniques for meeting these objectives compared. The technique which does the job effectively at the least cost should then be chosen. When viewed from this standpoint, the use of simulation as a study tool often becomes much more attractive economically.

3. The results of simulation must be accessible in a format that is meaningful to those using them. The actual simulation takes place within the computer and is, of course, unobservable to the user (neglecting the use of

some type of on-line visual display device). Thus it is necessary to devise some means of displaying simulation results in a form convenient to the user.

If each of the above considerations is satisfied, it is presumed that the simulation model is a useful tool in the study of signal timing.

Mode of Representation

There are various methods which may be used to represent the flow of traffic within the computer. Early traffic simulations employed a physical notation (11,16). Binary "1's" were used to represent vehicles and "0's" were used to indicate the spaces between vehicles. Groups of memory cells were figuratively placed end to end to represent the roadway. Algebraic manipulations caused the "1's" to change position, thereby simulating the flow of traffic. With this mode of representation the vehicles can occupy only certain specified locations (bit positions) along the roadway and individual vehicles have no identity as such.

The memorandum notation utilizes an entire word to represent a vehicle. Various parts of the word are used for such individual characteristics as its time of entry into the system, its desired velocity, et cetera. These parts may be extracted and interpreted as desired. This method is more versatile in that each vehicle's characteristics are identifiable as it moves through the

network. This makes it possible to compute the delays associated with the individual vehicle.

Most simulation programs which have used the memorandum notation have considered the roadway as being composed of a series of unit blocks (40). These blocks represent the various positions which a vehicle can occupy. Each block is one lane wide and has a length which is equivalent to some fraction or multiple of the unit vehicle length. Thus a vehicle may occupy only a limited number of discrete positions. Velocity and acceleration are step functions of the unit block and the time increment of scanning. This procedure is adequate for some models, but offers severe restrictions when realistic total delays are desired.

A third method of representation has been called a mathematical notation (11). This form of representation is similar to the memorandum notation except that, in addition to its other characteristics, each vehicle is associated with its own position indicator. Its position is therefore continuous within the accuracy of the computer. At any time a vehicle's new position can be computed as a function of its last position, its velocity, its acceleration, and the time increment. Spacings between vehicles are available from their respective coordinates and the vehicle length.

A fully mathematical notation has generally been avoided in the past since it requires a more complicated logic. Maneuvers, such as turns, which must be accomplished at a specified location were deemed more difficult when the vehicle could occupy any position at the start of the maneuver. Furthermore, the mathematical processing of the vehicles is more complex, thereby increasing the computer time required. On the other hand, the elimination of limitations on the position increment would allow some increase in the size of the time increment for the same model accuracy, and would provide increased versatility, thereby easing the further application of the model.

The SIGNET[1] model employs a fully mathematical notation. It was felt that the advantages to be realized from a virtually continuous position vector outweighted the additional execution time required. Furthermore, the high speed of the CDC 6500 computer somewhat offset this loss.

Each vehicle in the SIGNET model, then, is completely represented by the information contained in four computer words. Current position, current velocity, and current acceleration are each individual words (POSN, VEL, and ACCEL respectively). The fourth word, ICAR, contains

---

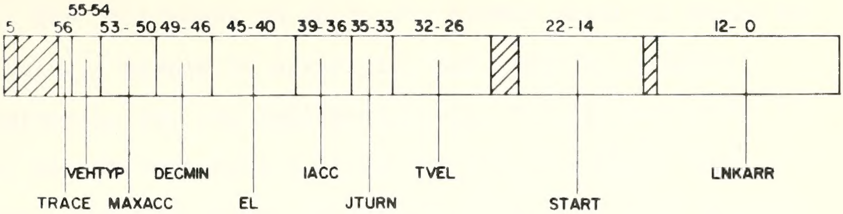
[1] SIGNET is the name of the simulation model developed in this project.

information on ten variables, as shown in Figure 1. The meaning and significance of each of these will be fully explained in later sections. Each variable is easily accessed via an unpacking function.

In actuality all of the drivers of vehicles within the roadway system are continually and simultaneously making decisions and modifying their behavior. The computer, however, can make only one simple logical choice at a time. In order to control all the occurrences at any given instant, it must process all decisions sequentially. In other words, it must process each decision for every vehicle in every lane, and for each lane in the system. It must do this in accordance with a prescribed sequence for each instance of time to be considered.

The selection of a suitable time increment is most important. If the time increment chosen is too large, it will not be possible to simulate all the events that may occur. On the other hand, if the increment is too small, many additional computations will be required for each event, resulting in additional computer time. In SIGNET the increment must be specified with the input.

In most simulations a critical factor is the minimum headway for vehicles. That is, since vehicles may enter the network only at each time increment, their minimum spacing will be equal to, or some multiple of, this increment. A method was developed for this study which isolated vehicle generation from the time increment.



LNKARR = link arrival time

START = initial coordinate at arrival on link

TVEL = target velocity

JTURN = turning movement

IACC = acceleration type

1 = stopped

2 = stopping

3 = stopped, waiting for turn

4 = stopping for turn

5 = stopping for lane change

6 = stopped for lane change

7 = free - behavior

8 = car following

9 = going in delayed left turn

10 = going in normal turn

11 = slowing for turn (no conflict)

12 = stopping for red signal

EL = effective length

DECMIN = minimum desired deceleration

MAXACC = maximum desired deceleration

VEHTYP = vehicle type

1 = car

2 = truck

TRACE = vehicle being traced on travel-time route

0 = no

1 = yes

FIGURE 1

CONTENT OF ICAR WORD

Mathematics of Vehicle Behavior

The philosophy on which the SIGNET model is based is relatively simple in principle, but involves complex programming for its implementation. The basic premise is that all drivers have a target velocity at which they would prefer to travel if conditions meet certain minimum requirements. Acting to limit the driver in the pursuit of his target velocity are obstructions generated by interactions with other vehicles and the physical environment. Limitations encountered include leading vehicles moving at a slower speed, red signal indications, turning movements, obstructions to lane changes, and conflicts with vehicles from other links at intersections.

The mathematical relationships describing vehicle behavior can be divided into 9 separate areas: vehicle generation, vehicle updating, car following, free behavior, amber acceptance, stopping performance, queue discharge, lane changing, and turning performance. These relationships are presented below in the order in which they might confront a vehicle being processed through the network.

Vehicle Generation. Vehicles are generated at the zero coordinate of each input link on a per-lane basis through the use of a translated negative-exponential distribution. Traditionally the unmodified negative exponential distribution has been used to obtain inter-vehicle headways. It is of the form

Mathematics of Vehicle Behavior

The philosophy on which the SIMTOT model is based is relatively simple in principle, but involves complex programming for its implementation. The basic premise is that all drivers have a target velocity at which they would prefer to travel if conditions were perfect without requirements. Subject to limits the driver, in the interest of his safety, ability and obstructions constrained by interactions with other vehicles and the physical environment. Limitations encountered include leading vehicles moving at a slower speed, red signal indications, turning movements, obstructions to lane changes, and conflicts with vehicles from other links at intersections.

The mathematical relationships describing vehicle behavior can be divided into 7 separate areas: vehicle generation, vehicle updating, car following, lane selection, speed conversion, stopping requirements, queue discharge, lane changing, and turning performance. These relationships are presented below in the order in which they would confront a vehicle being processed through the network.

Vehicle Generation. Vehicles are generated at the zero coordinate of each input link on a per-lane basis through the use of a translated negative-exponential distribution. Traditionally the unmodified negative exponential distribution has been used to obtain inter-vehicle headways. It is of the form

$$P(h \geq t) = e^{-\beta_1 t}$$

where

$P(h \geq t)$ = probability of headway being greater than or equal to t; and

β_1 = vehicle flow rate (veh./sec.).

However, being distributed in $(0,\infty)$, the negative-exponential does not compensate for a minimum headway which of course exists for every vehicle. Therefore, as proposed by Gerlough (34), a better approximation would be a negative-exponential with a translated axis:

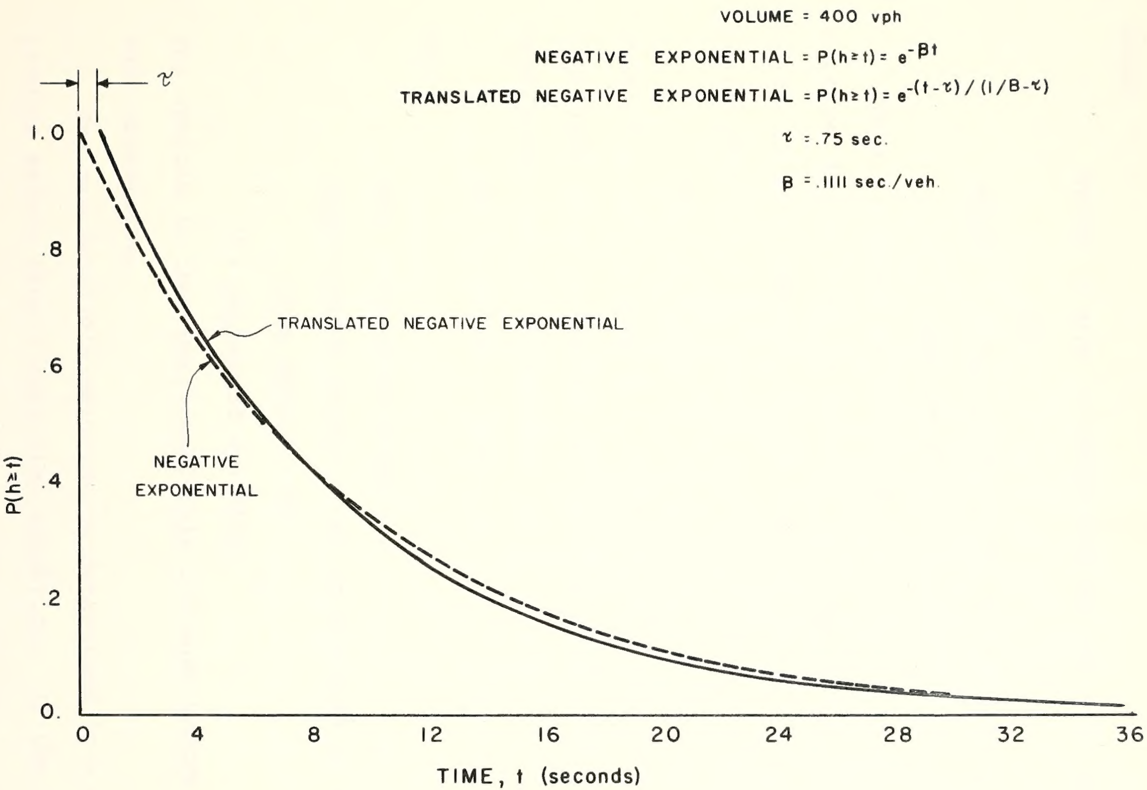
$$P(h \geq t) = e^{\frac{-(t - \tau)}{(1/\beta_1 - \tau)}}$$

where

τ = amount of translation; equivalently, in a physical sense, the minimum headway. (See Figure 2)

Work by Dawson and Chimini (10) in fitting the hyperlang probability distribution to intervehicular headways indicates a good value of τ (δ_1 in their study) to be 0.75 second. This value would apply for all flow rates.

In the SIGNET model the generation time is independent of the simulation scan cycle. For each lane in the network a variable is maintained to indicate the next arrival time of a vehicle on that lane. This tally is updated by the translated negative-exponential distribution at generation time to indicate the exact arrival time of the next vehicle in that lane.



VOLUME = 400 vph

NEGATIVE EXPONENTIAL = $P(h \geq t) = e^{-\beta t}$

TRANSLATED NEGATIVE EXPONENTIAL = $P(h \geq t) = e^{-(t-\tau)/(1/B-\tau)}$

τ = .75 sec.

β = .1111 sec./veh.

FIGURE 2

COMPARISON OF NEGATIVE EXPONENTIAL AND
TRANSLATED NEGATIVE EXPONENTIAL

TNEXT = TNEXT + NEGEXP

where

TNEXT = time of generation of the next vehicle; and

NEGEXP = headway determined by translated negative-exponential distribution.

Once a vehicle is generated its position is determined relative to the link coordinates. If the rear of its leader is at a coordinate less than zero, the vehicle is positioned a safe distance behind it:

POSN(J) = POSN(J-1) - EL(J-1) - 20.

Otherwise the distance the vehicle will travel at its generated velocity between the exact current time and its generation time is calculated:

DX1 = (TNEXT - TIME) · V

where

DX1 = distance traveled;

TIME = exact current time (a multiple of the scan cycle); and

V = generated velocity.

The vehicle is then positioned this distance before the zero coordinate.

The new vehicle's velocity is determined in a way to provide safety from a collision condition. If the spacing between it and its leader is less than 100 feet, its velocity is a function of both the spacing and the relative velocities:

$$DX2 = POSN(J-1) - EL(J-1) - POSN(J)$$

$$VEL(J) = VEL(J-1) + \frac{DX}{100} \cdot (TVEL(J) - VEL(J-1))$$

where

DX2 = intervehicular spacing; and

TVEL(J) = target velocity of the new vehicle.

Otherwise the velocity is set equal to the newly generated target velocity. Under normal volume conditions the previous model will receive little, if any, use. It is provided primarily to insure against collision conditions.

Several other descriptors of the vehicle's behavior are set at this time, some stochastically and others deterministically. The vehicle's target velocity (TVEL), minimum desired deceleration (DECMIN), maximum desired acceleration (ACCMAX), and effective length (EL) are generated probabilistically according to their respective distributions. Additionally the vehicle type (IVEHTYP) is randomly determined from the link truck percentage, input with the link traffic volume. The turning movement (JTURN) to be pursued at the link head is also determined randomly from the link turning probabilities. Finally, the link arrival time (LNKARR) is set to the current time, and the acceleration type (ACCTYP) is set to free behavior. All of the above are then packed into the ICAR word which was previously described (Figure 1.).

Vehicle Updating. The various behavior relationships to be described below yield only a negative or positive acceleration rate which begins after some reaction lag and continues for the rest of the scan cycle. In SIGNET both the reaction lag (REACT) and the scan cycle (CYCLE) are specified with the input data.

A theoretical analysis of driver reactions and highway events indicates the importance of driver reaction in safety and highway design. Therefore the inclusion of this parameter is important in achieving realism with the model. In practice a large variety of values are used for reaction time; the Traffic Engineering Manual (24), however, recommends a time between 0.75 and 1.0 second for design purposes in urban traffic. To maintain continuity with the Carstens data (8), used in validating the queue-discharge model, a time of 0.75 second is recommended for use in SIGNET.

Movements of vehicles between scans are computed by adaptations of the equations of motion:

$$\mathrm{V2} = \mathrm{VEL(J,I)} + \mathrm{ACCEL(J,I)} \cdot \mathrm{REACT}$$

$$\begin{aligned}\mathrm{POSN(J,I{+}CYCLE)} = {} & \mathrm{POSN(J,I)} + \mathrm{VEL(J,I)} \cdot \mathrm{REACT} + .5 \\ & \cdot \mathrm{ACCEL(J,I)} \cdot \mathrm{REACT}^2 + \mathrm{V2} \cdot \\ & (\mathrm{CYCLE} - \mathrm{REACT}) + .5 \cdot \mathrm{ACCEL(J,I+} \\ & \mathrm{REACT)} \cdot (\mathrm{CYCLE} - \mathrm{REACT})^2\end{aligned}$$

$$\mathrm{VEL(J,I{+}CYCLE)} = \mathrm{V2} + \mathrm{ACCEL(J,I{+}REACT)} \cdot \mathrm{(CYCLE{-}REACT)}$$

where

V2 = the velocity of vehicle J after the reaction period;

REACT = the reaction time;

CYCLE = scan cycle;

POSN(J,I) and
POSN(J,I+CYCLE)= the position of vehicle J at time I and I+CYCLE, respectively;

VEL(J,I) and
VEL(J,I+CYCLE) = the velocity of vehicle J at time I and I+CYCLE, respectively; and

ACCEL(J,I) and
ACCEL(J,I+REACT)= acceleration of vehicle J at time I and I+REACT, respectively.

Car Following. At the heart of the simulation model are the free-behavior and car-following relationships. Much of the full spectrum of behavior at an intersection involves a tracking or following process. An especially important example of this behavior is seen in the case of queue discharge. Therefore the stimulus-response equations of car following theory which were developed by Herman and associates (21,22) have been used in this study to describe certain patterns of intersection performance.

Herman's works pertaining to car following theory are highly valuable and apply directly to the problem of processing vehicles in a digital simulation. With two basic exceptions (turning movements and stopping), one of

the alternative vehicle behavior equations can be used to describe the behavior of individual vehicles within an intersection system.

Herman's equations have the general form

response = sensitivity x stimulus .

The best specific equation of this form Herman found was (in the notation of this study):

$$\text{ACCEL}(J+1,I+T) = a_0 \frac{\text{VEL}(J,I) - \text{VEL}(J+1,I)}{\text{POSN}(J,I) - \text{POSN}(J+1,I)}$$

where

ACCEL(J+1,I+T) = acceleration of car J+1, the follower, initiated at time I+T;

T = the car/driver lag;

VEL(J,I) and VEL(J+1,I) = the velocities of the leader and follower, initiated at time I;

POSN(J,I) and POSN(J+1,I) = the positions of the leader and follower initiated at time I; and

a_o = the characteristic speed.

This equation is termed the reciprocal spacing model.

A steady state equation which fits empirical data quite well takes the form

$$q = a_o K \ln(K_j/K)$$

where

q = flow;

a_o = characteristic speed;

K = concentration; and

K_j = jam concentration.

This relationship has been successfully applied to freeway and tunnel flow successfully in the past, but under a specific intersection condition the equation is especially pertinent. This condition occurs during that portion of the green phase when flow is maximum. By differentiating and setting $dq/dK = 0$, it is found that when q approaches q_{max}, K_j/K approaches e. Thus by choosing typical values for K_j, related values of q_{max} and a_o can be computed. (It should be noted that q_{max} is analogous to saturation flow; in fact, by definition, these two values are probably the same thing for a line of straight-through vehicles.)

This relationship is directly related to this work due to its compatibility with the model of individual vehicle behavior, the reciprocal spacing model. The characteristic speed, a_o, in the steady-state relationship is identical to that in the car following equation. Consequently, for any intersection it would only be necessary to measure maximum flow past the entry point (for a series of straight through movements from a long queue). Then a_o could then be calculated and used in the

car following equation. It should be understood that a_o does not correspond to the target velocity around which free-flowing vehicles center, but is the typical speed past the entry point of vehicles that started as part of a long stationary queue once maximum flow has been attained. Generally speaking, passage through the green of undelayed free-flowing vehicles will not result in maximum flow.

Free Behavior. Not all vehicles in a real system act as followers, however. An example of such a vehicle is the leader of a queue being discharged from a signal. In such a case behavior can be described by

$$\text{ACCEL}(J,I+T) = K\ (\text{TVEL}(J) - \text{VEL}(J,I))$$

where

ACCEL(J,I+T) = acceleration of car J initiated at time I+T;

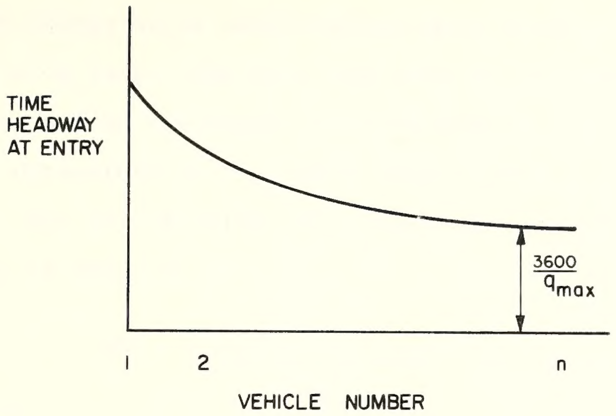
K = proportionality coefficient;

TVEL(J) = target velocity of car J; and

VEL(J,I) = velocity of car J at time I.

This equation is termed the free behavior model. Applying the reciprocal spacing model to queue members and the free behavior model to queue leaders should result in the general relationships shown in Figure 3.

Response to Amber Signal. The decision to stop or go on amber is based on probability distributions developed for the TRANS model (14). Data were taken from the work of Olson and Rothery (29) and Blackman and Crawford (6).



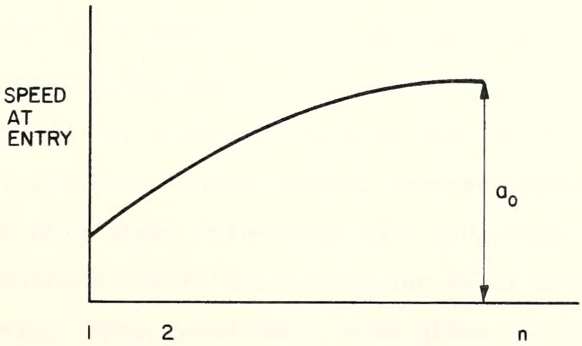


FIGURE 3

GENERAL QUEUE DISCHARGE RELATIONSHIPS

Results of these studies, relating probability of stopping to either distance or travel time from the intersection, were converted to deceleration rates required to stop at the stop line. The data were plotted on normal probability paper and an excellent linear regression fit (R = 0.967) was determined by the probit method (Figure 4).

The deceleration rate needed to stop at the stop line is computed by

$$\text{ACC} = \frac{\text{VEL(J)}^2}{2\ (\text{STOP} - \text{POSN(J)})}$$

where

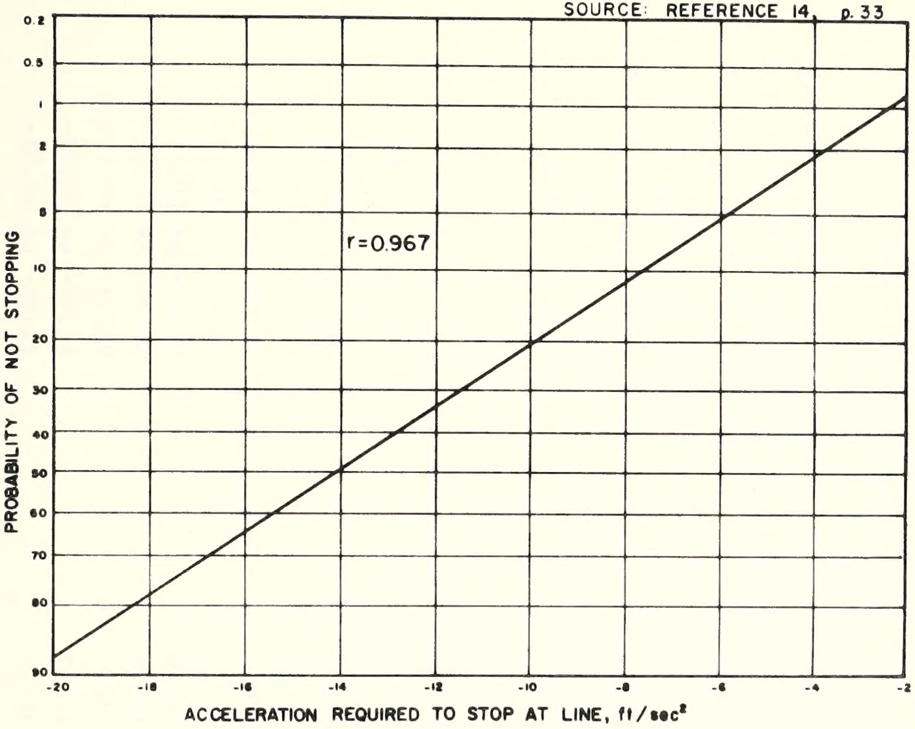
ACC = required negative acceleration; and

STOP = target stopped position.

The decision to stop or go is made probabilistically by reference to an amber decision probability table (Table 1).

Stopping Performance. Two types of stops occur in the model: (1) stopping first in line at the intersection, and (2) stopping behind another stopped vehicle. Empirical work at Ohio State University (35) indicated that the use of a constant deceleration stopping model would be quite realistic. They found that, when given the choice, drivers tended to decelerate to a stop at an approximately constant rate.

In the SIGNET stopping model, the parameters of a minimum desired deceleration rate are supplied and a value DECMIN(J) is randomly selected for each vehicle J at its



SOURCE: REFERENCE 14, p. 33

FIGURE 4

AMBER DECISION PROBABILITY DISTRIBUTION

TABLE 1

Amber Decision Probability

Acceleration Required to Stop at Stop Line (ft/sec^2)	Probability of Stopping
-0 to -0.99	1.000
-1.00 to -1.99	0.994
-2.00 to -2.99	0.989
-3.00 to -3.99	0.982
-4.00 to -4.99	0.972
-5.00 to -5.99	0.956
-6.00 to -6.99	0.935
-7.00 to -7.99	0.905
-8.00 to -8.99	0.867
-9.00 to -9.99	0.820
-10.00 to -10.99	0.762
-11.00 to -11.99	0.700
-12.00 to -12.99	0.624
-13.00 to -13.99	0.548
-14.00 to -14.99	0.468
-15.00 to -15.99	0.390
-16.00 to -16.99	0.318
-17.00 to -17.99	0.250
-18.00 to -18.99	0.190
-19.00 to -19.99	0.140

generation time. For each scanning cycle the required stopping rate for the first vehicle on the approach is computed:

$$\mathrm{ACCEL(J,I)} = \frac{\mathrm{VEL(J,I)}^2}{2\ (\mathrm{POSN(J,I)} - (\mathrm{STOP} + \mathrm{E1}))}$$

where

STOP = the position of the stop line; and

E1 = random normal stopping error.

The other terms have been defined previously.

When ACCEL(J,I) $\leq$ DECMIN(J), the vehicle begins stopping at rate ACCEL(J,I) and continues to do so until zero velocity is reached. This approach tends to give the model added stability.

A similar model is used for stopping behind another vehicle, with the main difference being in the computation of the target stopped position. Vehicles stop at the position of the effective rear of the previously stopped vehicle.

The stopping model is not used until the leader has begun to stop. At that time the projected stopped position of the leader is computed:

$$\mathrm{PSL} = \frac{-\ \mathrm{VEL(J\text{-}1,I)}^2}{2\cdot\mathrm{ACCEL(J\text{-}1,I)}} + \mathrm{POSN(J\text{-}1,I)}$$

where

PSL = projected stopped position of leader ahead of vehicle J;

VEL(J-1,I) = current velocity of leader ahead of vehicle J;

ACCEL(J-1,I) = current stopping acceleration of leader ahead of vehicle J; and

POSN(J-1,I) = current position of leader ahead of vehicle J.

The target stopped position of the vehicle under consideration is then found by:

PS = PSL - EL(J-1)

where

PS = target stopped position of vehicle J; and

EL(J-1) = effective length of leader ahead of vehicle J.

The effective length of a vehicle is equivalent to the average stopped spacing of vehicles stopped in queue. This average stopped spacing is measured from the front bumper of the leading vehicle to the front bumper of the following vehicle, and therefore includes the vehicle length and a clear space. Field studies have shown that EL has a value of approximately 22 feet for cars (3, 18, 26, 34). In SIGNET a vehicle's effective length is determined probabilistically at generation time from the parameters supplied as input.

The stopping acceleration is found similarly to the case of the first vehicle on the approach, discussed above. A secondary check is then performed if the stopping acceleration is rejected, in which the projected stopping acceleration at the next scan cycle is calculated:

$$ACC = \frac{VEL(J,I+C)^2}{2\ (POSN(J,I+C) - PS)}$$

where

ACC = projected stopping acceleration at time I+C;

C = time interval;

VEL(J,I+C) = projected velocity of vehicle J at time I+C;

POSN(J,I+C) = projected position of vehicle J at time I+C; and

PS = target stopped position of vehicle (same value as was computed previously).

Studies have shown that deceleration rates of 8 to 9 feet per second are comfortable while rates up to 16 feet per second per second can be used without severe discomfort to the driver (19,39). For use in this stopping model a maximum desired deceleration rate of 10 feet per second per second was chosen. The projected stopping acceleration rate calculated above is compared to this value; if it is more severe, the previously calculated stopping acceleration is accepted and the vehicle begins

to stop. Otherwise it continues with zero acceleration.

Queue Discharge Performance. As in the stopping model, two types of vehicle action occur during queue discharge: (1) starting from first in line at the intersection, and (2) starting behind another vehicle. The first vehicle in line uses a modification of the free-behavior equation

$$\text{ACCEL}(J,I+T) = K\ (\text{TVEL}(J) - \text{VEL}(J,I))$$

The terms of the equation have been defined previously.

Work by Carstens (8) in the area of queue discharge revealed the average starting delay for the first vehicle in queue to be 0.35 seconds. Accordingly this value is used for T (the car/driver lag) in the above free behavior equation. When the first vehicle discharging from the queue is not being considered or when the vehicle is already underway T reverts to the original value found in the input data.

The original starting model for following vehicles used a simple application of the reciprocal spacing model

$$\text{ACCEL}(J+1,I+T) = a_o \frac{\text{VEL}(J,I) - \text{VEL}(J+1,I)}{\text{POSN}(J,I) - \text{POSN}(J+1,I)}$$

This version of the model was very dependent on the simulation time interval, however, since the state of the queue could only be checked once every scan cycle.

In reality, the driver is continuously analyzing his relationship to his leader. To approximate this behavior in the queue discharge model the leader-follower relationship is examined 1 second after the leader beings moving, and the follower's movement is begun. In this way a 1 second delay results which would adequately account for the car/driver lag, and the starting model is kept independent of the simulation time interval.

Independent studies by Greenshields (17,18), Carstens (8), and Bartle et al (2) have resulted in estimates of both starting delay and succeeding time spacings between vehicles discharging from a queue (Table 2). The variation in these estimates can be at least partially accounted for by the known influence of intersection geometrics on capacity. The SIGNET starting model, shown in Figure 5 with 95% confidence intervals for 50 iterations, shows good agreement with the other studies, even though confidence intervals were not reported with them.

Lane Changing. Lane change maneuvers associated with passing are not allowed in the SIGNET model. This restriction is not unrealistic since it is presumed that in most urban situations traffic volumes are high enough to greatly reduce the desire for such maneuvers. Lane changes associated with turning movements are permitted, however, since quite often the lane from which a vehicle wishes to turn is different from that in which it entered the link.

TABLE 2

Queue Discharge - - Seconds to Cross Stop Line

Vehicle Number	Greenshields Data	Bartle Data	Carstens Data	SIGNET Model	Confidence Interval
1	3.8	(3.1)	2.64	1.03	1.01 - 1.05
2	6.9	5.8	5.13	4.66	4.58 - 4.75
3	9.6	7.9	7.62	7.66	7.54 - 7.77
4	12.0	10.0	9.91	9.90	9.74 - 10.07
5	14.2	12.1	12.20	12.68	12.56 - 12.81
6	16.3	14.2	14.49	14.95	14.79 - 15.11
7	18.4	16.3	16.78	17.30	17.08 - 17.52
8	20.5	18.4	19.07	19.58	19.34 - 19.82
9	22.6	20.5	21.36	21.84	21.52 - 22.15
10	24.7	22.6	23.65	24.10	23.67 - 24.52

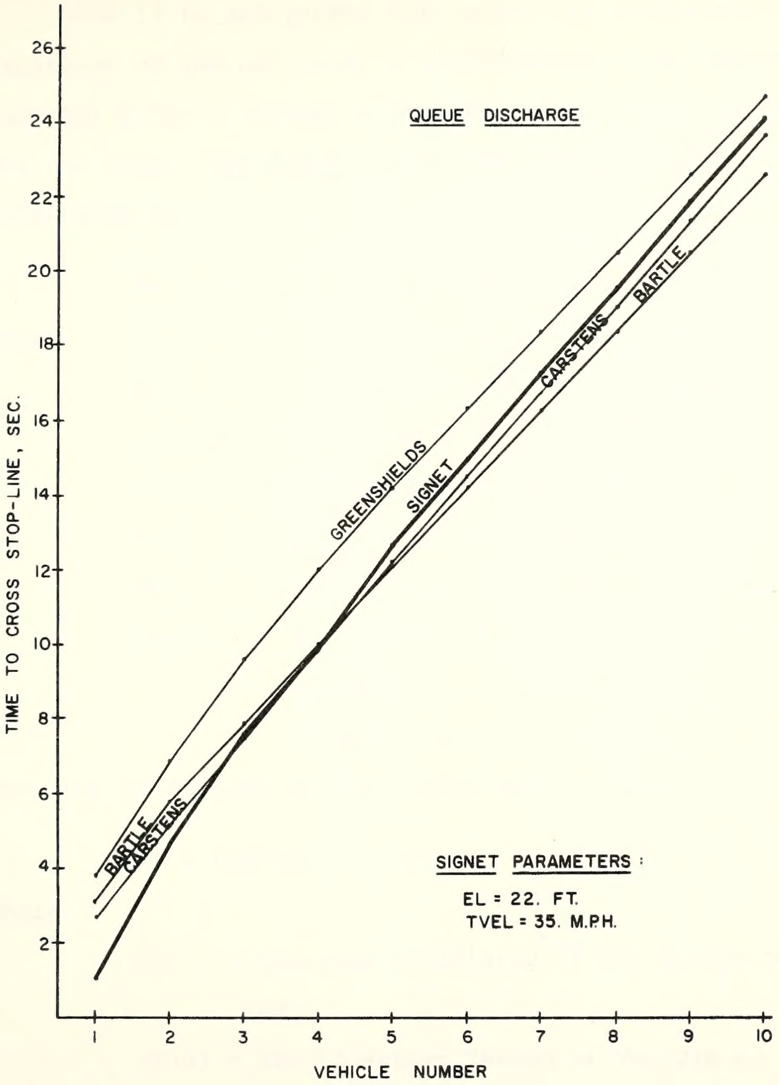


FIGURE 5

QUEUE DISCHARGE COMPARISON

Once it is determined that geometric conditions (i.e. existence of the new lane, a consideration with turning pockets) allow a change, a changing zone is defined in the new lane. The front coordinate of this zone is calculated by:

$$XF = POSN(J) + \overline{EL} + SPACE$$

where

XF = the front coordinate of the change zone;

POSN(J) = the current position of vehicle J, the vehicle under consideration;

$\overline{EL}$ = the average effective length; and

SPACE = the minimum allowable spacing, equal to one half of the vehicle's velocity (in m.p.h.).

The rear coordinate of the change zone is defined by:

$$XR = POSN(J) - EL(J)$$

where

XR = the rear coordinate of the change zone; and

EL(J) = the effective length of vehicle J. (See Figure 6).

If another vehicle is in this change zone, the change is rejected.

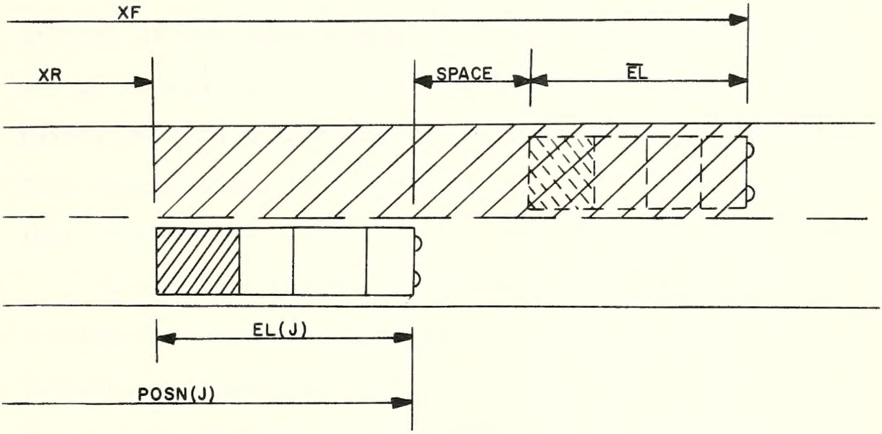


FIGURE 6.
CHANGE ZONE DEFINITION

In the absence of such conflict the potential lane changer determines the expected travel time to the change zone of the next vehicle in the new lane, based on its current position and velocity. This travel time is defined as the lane change lag. The decision to change or not change is then made probabilistically by reference to a lane change lag acceptance table, Table 3. This distribution is taken from the TRANS model (14). (See Figure 7).

If the lane change is rejected the vehicle begins slowing at a rate equal to twice the vehicle's minimum desired acceleration. Occasionally, especially under high volume conditions, a hangup could result when two potential lane changers stop in each other's change zone. To avoid this, the vehicle in the left lane adjusts its stopping rate to stop behind the other's change zone through the following series of computations:

$$\text{PFADJ} = \text{POSN}(K) - \frac{\text{VEL}(K)^2}{2\ (\text{ACCEL}(K))}$$

$$\text{PFLZN} = \text{PFADJ} - 10. - \text{EL}(K)$$

$$\text{ACCEL}(J) = -\frac{\text{VEL}(J)^2}{2\ (\text{PFLZN} - \text{POSN}(J))}$$

where

J = vehicle in left lane;

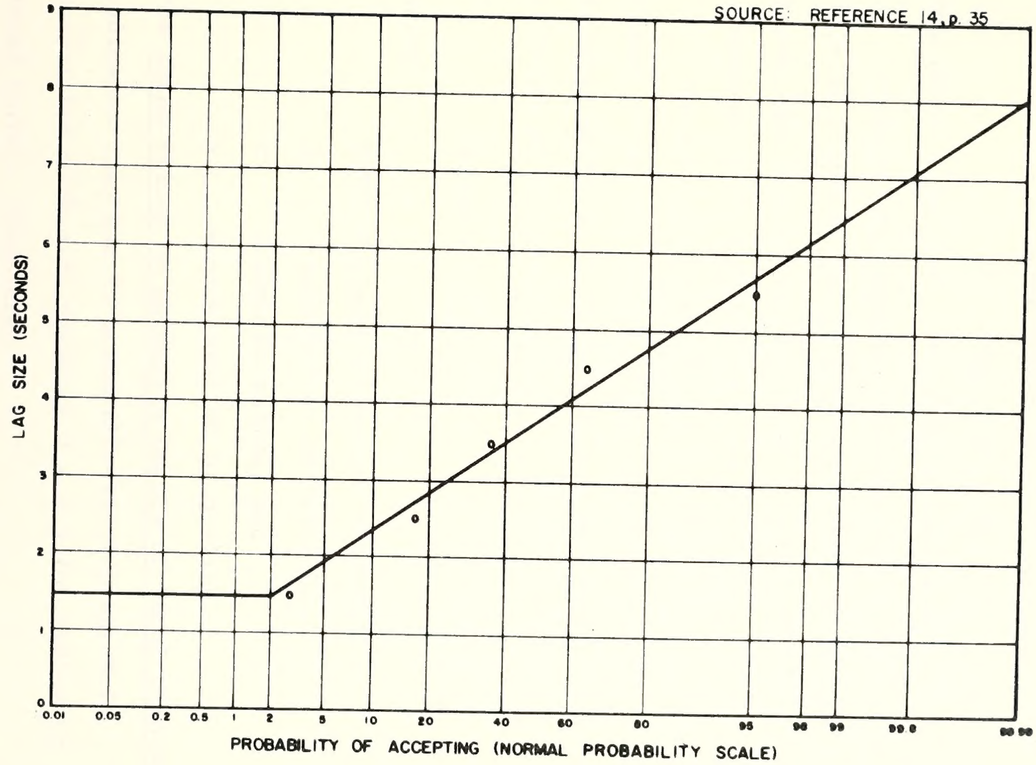
K = conflicting vehicle in right lane;

PFADJ = projected final position of vehicle K; and

TABLE 3

Lane Change Decision Probability

Lane Change Lag (Sec.)	Probability of Acceptance
0 to 0.5	0
0.5 to 1.0	0
1.0 to 1.5	0
1.5 to 2.0	0
2.0 to 2.5	0.090
2.5 to 3.0	0.180
3.0 to 3.5	0.310
3.5 to 4.0	0.490
4.0 to 4.5	0.660
4.5 to 5.0	0.810
5.0 to 5.5	0.900
5.5 to 6.0	0.960
6.0 to 6.5	0.985
6.5 to 7.0	0.995
7.0 to 7.5	0.999
7.5 to 8.0	1.000
8.0 to 8.5	1.000
8.5 to 9.0	1.000
9.0 to 9.5	1.000
9.5 to 10.0	1.000



SOURCE: REFERENCE 14, p. 35

FIGURE 7

LANE CHANGE DECISION PROBABILITY DISTRIBUTION

PFLZN = projected final rear coordinate of vehicle K's change zone; equivalently, the target stopped position of vehicle J. (See Figure 8).

The other terms have been defined previously.

Turning Performance. Vehicles which desire to turn left or right at the intersection must at some point cease operating under the stimulus-response model and undertake an independent fixed turning schedule.

The principal requirement is that vehicles must not exceed a certain maximum speed during the turn. Maximum turning velocity is related to turning radius and side friction by the equation

$$\text{VTURN} = \sqrt{fgr}$$

where

VTURN = maximum turning, ft./sec.;
f = coefficient of friction;
r = turning radius; and
g = acceleration of gravity.

The AASHO Policy on Geometric Design of Rural Highways (1) indicates that the 95-percentile turning speed is associated with side friction f = 0.3 for medium to low-speed turns. Therefore,

$$\text{VTURN} = \sqrt{0.3\ (32.2)\ r} = \sqrt{9.66\ r}$$

where VTURN is in feet per second and r is in feet. Thus

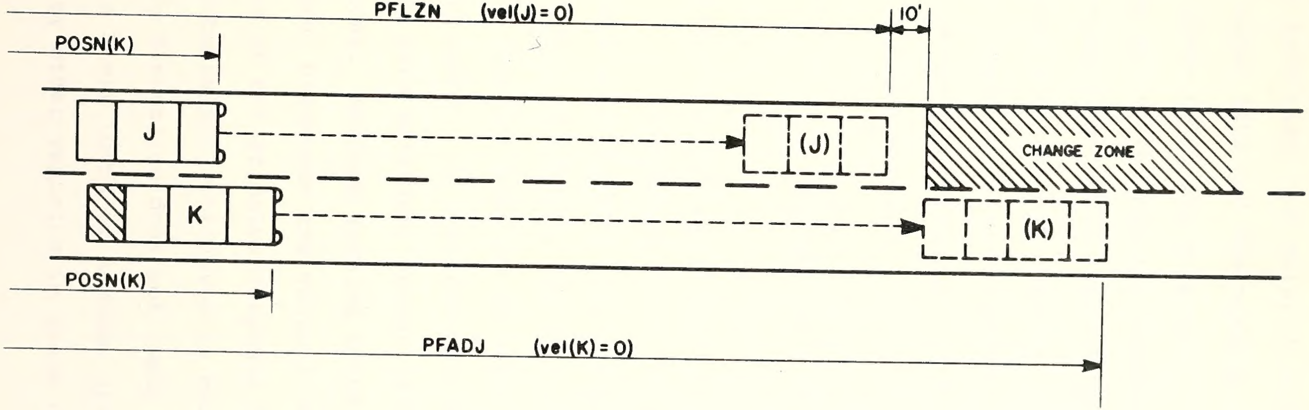


FIGURE 8

VARIABLES USED IN COMPUTING ADJUSTED
STOPPING RATE

right-turn, free left-turn, and delayed left-turn radii are supplied as input data and the maximum speeds associated with these radii are computed.

As a turning car approaches the intersection it is scanned at each simulation cycle, and if the current velocity is greater than maximum turning velocity, the deceleration rate required to reach maximum turning velocity exactly at the start-turn point is computed. For example, for a right turner,

$$\text{ACCEL(J,I+T)} = \frac{\text{VTURN}^2 - \text{VEL(J,I+T)}^2}{2\ (\text{BEGINPT(RT)} - \text{POSN(J,I+T)})}$$

where

BEGINPT(RT) = start right turn position; and

VTURN = maximum right-turn velocity.

The other terms have been defined previously.

When ACCEL(J,I) $\leq$ DECMIN(J), the vehicle begins slowing to maximum turning velocity at rate ACCEL(J,I), as in the stopping model. It continues to do so until reaching the start turn position, unless affected by more stringent conditions. Maximum turning speed is maintained throughout the turn, whereupon the vehicle resumes behavior under one of the stimulus-response models (car-following or free-behavior), whichever is more stringent.

The foregoing discussion does not imply that all vehicles will make their turns at maximum speed. Some will be affected by other vehicles in queue or vehicles

on the receiving link so that their turns will be made at considerably lower speeds. However, none will exceed maximum turning speed.

Left-turns are subject to further complications because they must be made across the path of opposing traffic. The driver must decide if gaps and lags in the opposing approach traffic are sufficiently large to be accepted. In this model, when a left-turning driver crosses the start-free-left-turn position, a decision is made to either accept or reject a time lag (i.e., the time required for the first straight-through vehicle on an opposing approach to reach the center of intersection conflict point). This lag is computed by

$$V = VEL(K)^2 + 2 \cdot ACCEL(L) \cdot (CENTER - POSN(K))$$

$$DT = \frac{(VEL(K) - V)}{ACCEL(K)}$$

where

K = opposing vehicle;
CENTER = coordinate of center of intersection;
V = projected velocity of vehicle K at CENTER; and
DT = time lag.

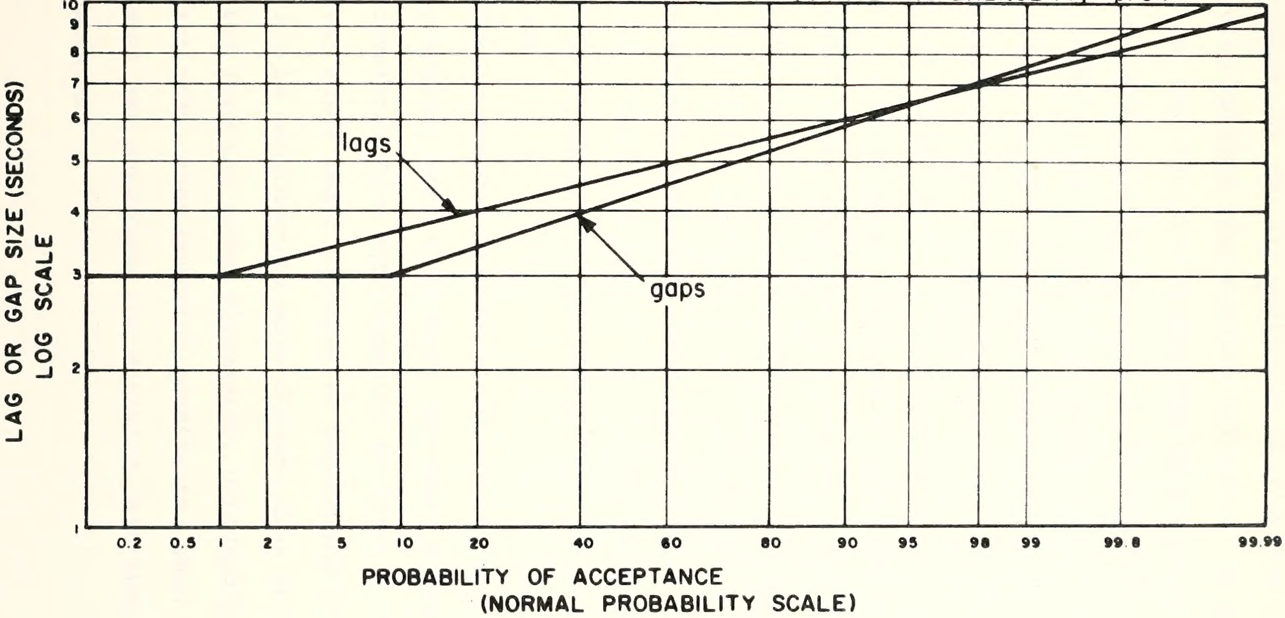
The other terms have been defined previously.

The decision to accept or reject the time lag is made probabilistically by reference to a left-turn lag acceptance table (Table 4 and Figure 9). If the lag is

TABLE 4

Left-Turn Decision Probability

Lag or Gap Size (Sec.)	Probability of Accepting Lag	Probability og Accepting Gap
0 to 0.5	0	0
0.5 to 1.0	0	0
1.0 to 1.5	0	0
1.5 to 2.0	0	0
2.0 to 2.5	0	0
2.5 to 3.0	0	0
3.0 to 3.5	0.030	0.150
3.5 to 4.0	0.124	0.320
4.0 to 4.5	0.300	0.520
4.5 to 5.0	0.530	0.690
5.0 to 5.5	0.730	0.820
5.5 to 6.0	0.860	0.900
6.0 to 6.5	0.940	0.950
6.5 to 7.0	0.970	0.970
7.0 to 7.5	0.990	0.986
7.5 to 8.0	0.996	0.993
8.0 to 8.5	0.999	0.997
8.5 to 9.0	1.000	0.998
9.0 to 9.5	1.000	0.999
9.5 to 10.0	1.000	1.000



SOURCE: REFERENCE 14, p. 34

FIGURE 9

LEFT-TURN DECISION PROBABILITY DISTRIBUTIONS

acceptable the vehicle continues through the turn, never exceeding maximum free-turning velocity. If the lag is rejected, the vehicle immediately begins stopping at the constant rate required to stop at the left-turn waiting point, one quarter of the way through the delayed turn. Thereafter, the driver views each time gap appearing in the opposing traffic and determines its acceptability by reference to the left-turn gap acceptance probability table. If acceptable, the vehicle negotiates a delayed left-turn, never exceeding maximum delayed-turning velocity. It is possible, due to the use of random numbers in the model structure, that a waiting vehicle will at some point accept a short gap which it would have previously rejected. Such behavior is not unrealistic, since in actuality a degree of impatience often bears upon such a decision.

The log-normal probability distributions for gap and lag acceptance were taken from the TRANS model (14). They were developed from data supplied by James H. Kell.

The exact logic employed in processing left-turn vehicles is considerably more complex than would be inferred from the above discussion, principally due to the fact that other behavior conditions (such as stopping in queue) may supercede the standard left-turn schedule and decisons.

In the above sections the basic mathematical relationships governing an individual vehicle's behavior were presented. The next chapter, Chapter III, is concerned with the logic which employs these relationships in processing a vehicle through the network.

CHAPTER III

WORKING DETAILS OF THE SIMULATION MODEL

The original purpose of SIGNET was to evaluate the effect of traffic signal settings on traffic flow in a region of a city. Thus SIGNET is a tool which helps to optimize the timing of traffic signals in a street network. Alternatively SIGNET may be used to study the effect of other network modifications including: geometric design changes, such as additional turning lanes; traffic regulation changes, such as turning movement and parking prohibitions; and changes in the magnitudes and patterns of traffic demands.

In the SIGNET model a traffic network is considered to consist of a set of links. Each link carries traffic in one direction between two signalized intersections. Links are of three types: input, network, and output. An input link feeds traffic into the network; a network link carries traffic from one node (i.e. signalized intersection) in the interior of the network to another; and an output link removes traffic from the network.

For identification purposes each link is assigned a positive integer. Input links are numbered

1,2,3,...,n_I; network links are numbered (n_I+1), (n_I+2), ...,(n_I+n_N); and output links are numbered (n_I+n_N+1), (n_I+n_N+2),..., $(n_I+n_N+n_0)$. Because a digital computer can perform only one operation at a time, it is necessary for the simulation program to process each vehicle sequentially. In the SIGNET model this sequence is ordered by link number, with the vehicles within the link being processed from the link head to tail.

Special care should be taken in numbering the links so that all links feed lower numbered links (i.e. links which have already been processed). This is quite easy to accomplish in open networks, such as single arterials. Closed networks are more difficult, however, and in most cases the number of links feeding higher numbered links can only be minimized. These cases should, wherever possible, be confined to low volume locations to minimize their effect. Additionally, the SIGNET model employs supplementary logic to make the processing of such cases more accurate.

Each link (input type or network type) has a traffic signal associated with it. The signal appears at the head of the link, and may take on any of the 14 indications shown in Table 5. Only one of these may be on at any given time.

The characteristics of each link in the network must be supplied to the program as input. Physical

TABLE 5

Signal Indications

Number	Indication	Movements Permitted
1	Red	None
2	Green	all
3	Green Arrow Straight	#3, #4
4	Green Arrow Left	#6
5	Green Arrow Right	#1
6	Green Arrow Diagonal Left	#5
7	Green Arrow Diagonal Right	#2
8	Green Arrow Straight, Green Arrow Left	#3, #4, #6
9	Green Arrow Straight, Green Arrow Right	#1, #3, #4
10	Green Arrow Straight, Green Arrow Diagonal Left	#3, #4, #5
11	Green Arrow Straight, Green Arrow Diagonal Right	#2, #3, #4
12	Green Arrow Left, Green Arrow Diagonal Left	#5, #6
13	Green Arrow Right Green Arrow Diagonal Right	#1, #2
14	Amber	all

characteristics are defined through the use of PRESIG, the preliminary geometry determination program, which converts easily obtained geometry data to the required inputs of SIGNET. A full description of this program follows. Other link inputs include probabilistic characteristics pertaining to arrival and departure of vehicles and traffic signal characteristics which define the timing for each of the signal modes. Additionally, vehicular arrival rates are required for input links.

A simulation run is divided into two major parts: fill time and observe time. During fill time traffic is generated on the input links and pumped into the network until a stable condition is reached; this serves to create an initial network population. Generally a 15 minute fill time should be adequate, although this should be somewhat dependent on network size and configuration and magnitude of traffic flow. At the end of fill time appropriate initializations are performed and observation begins.

Simulation is carried out in discrete time intervals. During each simulation cycle four major operations are performed: (1) vehicles are generated on the input links; (2) vehicles are advanced within each link as conditions allow; (3) vehicles are moved from the front of their link to the rear of the receiving link at signalized intersections; and (4) bookkeeping operations are

performed. The bookkeeping operation records or summarizes all movements and delays. These data provide the various statistics that are of significance to the traffic engineer, such as vehicle-miles travelled, average attained speeds, densities, and volumes.

A variety of options are available in the use of SIGNET. For purposes of travel time testing or for more detailed analysis, a route may be designated through the network. A frequency analysis of travel times is then printed for each link on this route whenever a normal summary occurs. Similarly, a frequency analysis of queue lengths may be specified on up to 15 links. Turning probabilities, link input volumes, and truck percentages may each be reset up to 10 times in the course of one run. Primarily for debugging purposes, a full printout of all vehicles' positions, velocities, accelerations, and other pertinent data may be requested for any period of time. Finally, up to 15 complete summaries of flow statistics may be specified at any time during the observation time.

Geometric Configuration

The roadway is not represented physically, as such, in the computer. However, it is necessary to specify certain roadway references in order to make meaningful the positions of vehicles contained in the POSN word. Thus there is a need for a coordinate system in which vehicles operate. Consider the sketch (Figure 10) of a

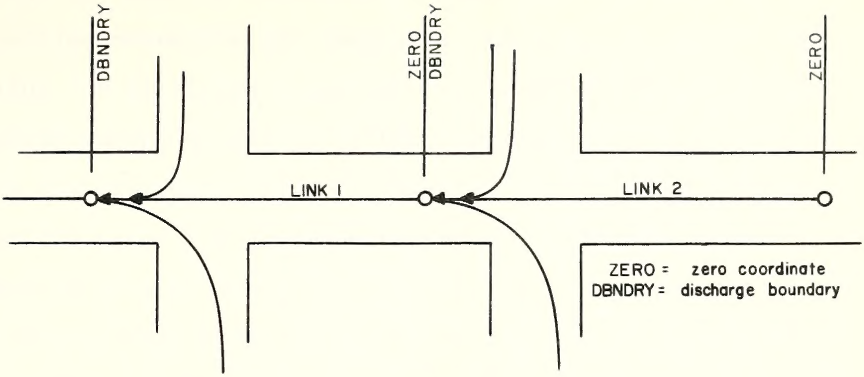


FIGURE 10
A SIMPLIFIED LINK PAIR

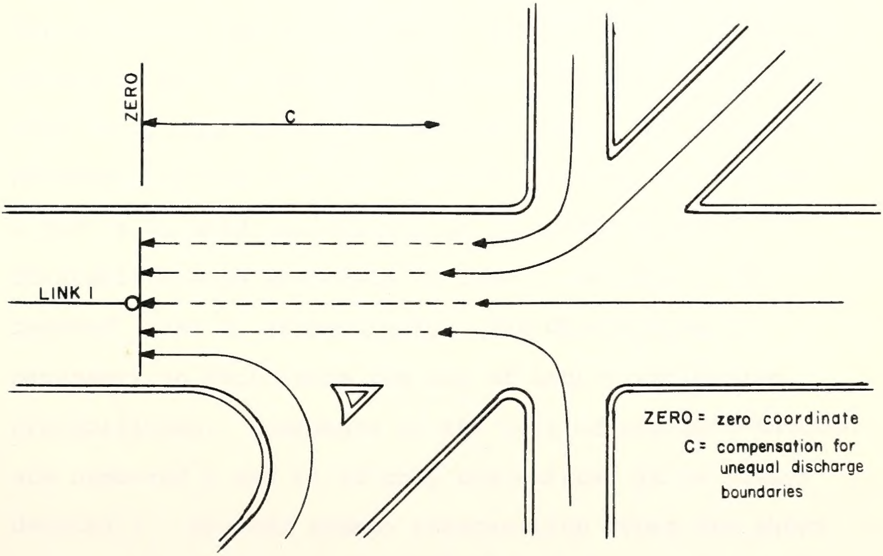


FIGURE 11
COMPENSATION FOR UNEQUAL DISCHARGE BOUNDARIES

simplified pair of links. The zero coordinate of each link is taken to be the point where all contributing turning movements are complete. In turn, the end of the link, or discharge boundary, is the point where all other contributing movements to the receiving link are complete. In reality there could be several contributing movements at the tail (i.e. zero coordinate) of the link, each ending at a different point. It would be necessary to figuratively add a tangent section to each except the longest, thereby making all end at the same point (Figure 11).

There are a total of 8 possible movements at the head of a link (Figure 12). Of these, any 6 may be used for each link within the structure of SIGNET. These six movements are numbered in a counter-clockwise order, as they would confront a vehicle approaching the intersection. Movements to the right of straight-through are denoted 1 and 2; if only one exists, it is always denoted 1. Straight-through movements in lane 1 and lane 2 are denoted 3 and 4, respectively. The distinction is necessary to facilitate the use of lane distribution probabilities. Movements to the left of straight-through are numbered 5 and 6; if only one exists, it is always denoted 6. Several common intersection types are shown in Figure 13 with movement numbers illustrating this convention.

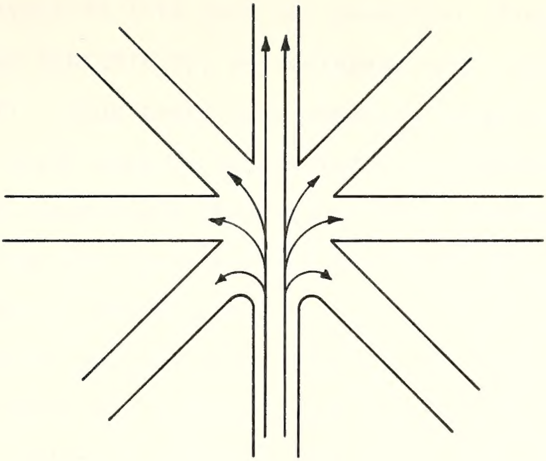


FIGURE 12

LEGAL TURNING MOVEMENTS

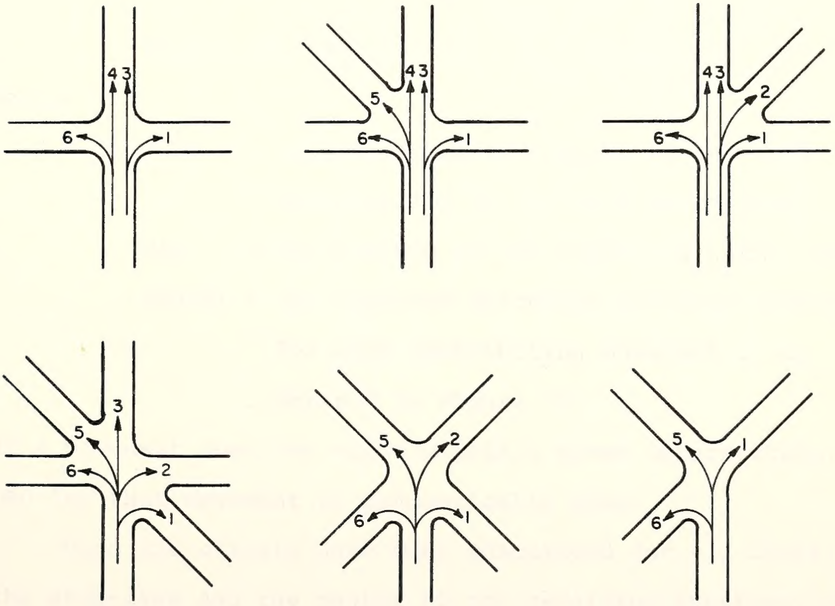


FIGURE 13

SAMPLES OF TURNING MOVEMENT COMBINATIONS

Two types of left turn are possible: free flowing, with radius RADIUS(FLT), and delayed, with radius RADIUS(DLT). Thus there nine possible contributing movements which must be considered in the determination of the zero coordinate (see Figure 14). Straight-through movements are not considered in this respect because their length is not critical.

The first step in tying down the network geometry is to determine for each link the location of the zero coordinate relative to the center of the contributing, or tail, intersection. Referring to Figures 11, 14, and 15, the following equation can be written:

$$\text{ZERO} = \text{MAX}\ (\text{DBO}(1), \text{DBO}(2), \ldots, \text{DBO}(9))$$

where

ZERO = distance from the center of the tail intersection to the zero coordinate;
MAX = the maximum of the following terms; and
DBO(I) = the required discharge boundary offset for each contributing movement I, as derived in Figure 15.

If a movement does not exist within a given intersection, DBO for that movement is automatically zero.

Once the offsets have been calculated for all links, the stop-line and the center of the receiving intersection may be redefined relative to the zero coordinate, leaving the way clear to compute the beginning of turn points and

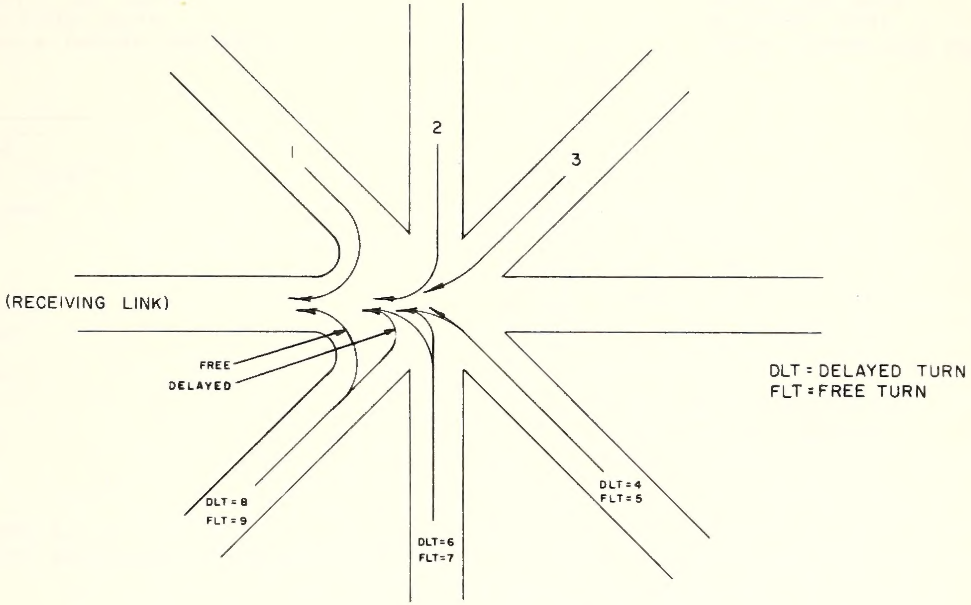
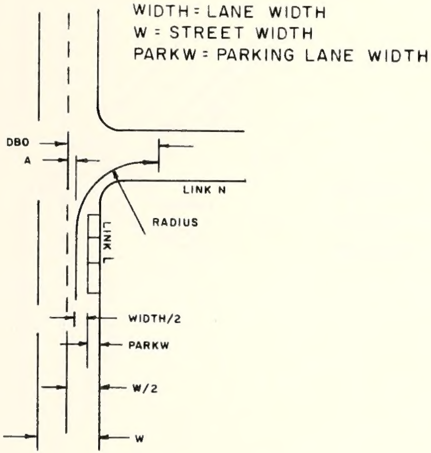


FIGURE 14.

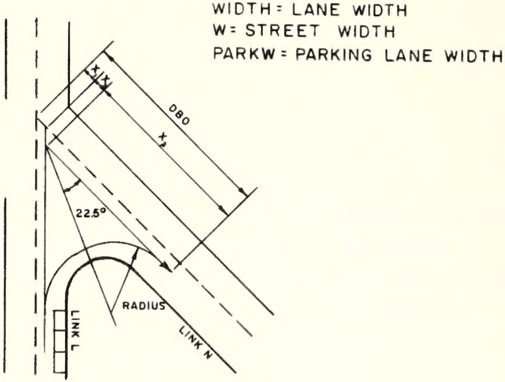
CONTRIBUTING TURNING MOVEMENTS



WIDTH = LANE WIDTH
W = STREET WIDTH
PARKW = PARKING LANE WIDTH

$$\begin{aligned} DBO(2) &= RADIUS + A \\ &= RADIUS + W(L)/2 - WIDTH/2 - PARKW(L) \end{aligned}$$

FIGURE 15a.



WIDTH = LANE WIDTH
W = STREET WIDTH
PARKW = PARKING LANE WIDTH

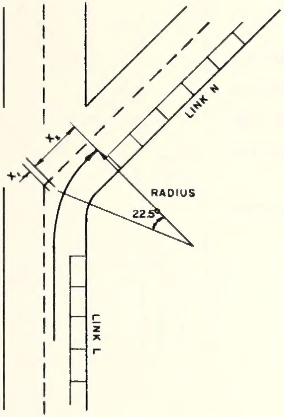
$$X_1 = (W(L)/2 - WIDTH/2 - PARKW(L))\sqrt{2}$$
$$X_2 = (W(N)/2 - WIDTH/2 - PARKW(N))$$
$$X_3 = RADIUS/TAN(22.5^\circ)$$

$$\begin{aligned} DBO(1) &= X_1 + X_2 + X_3 \\ &= (W(L)/2 - WIDTH/2 - PARKW(L)) * \sqrt{2} \\ &+ (W(N)/2 - WIDTH/2 - PARKW(N)) \\ &+ RADIUS/TAN(22.5^\circ) \end{aligned}$$

FIGURE 15b.

FIGURE 15.

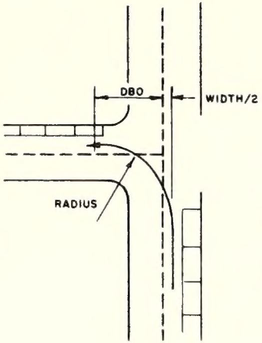
COMPUTATION OF DISCHARGE BOUNDARY OFFSET



WIDTH = LANE WIDTH
W = STREET WIDTH
PARKW = PARKING LANE WIDTH

$$\begin{aligned} DBO(3) &= X_1 + X_2 \\ &= RADIUS(TAN(22.5^\circ)) \\ &\quad -(W(N)/2 - WIDTH/2 - PARKW(N)/2)\,TAN(22.5^\circ) \end{aligned}$$

FIGURE 15c.



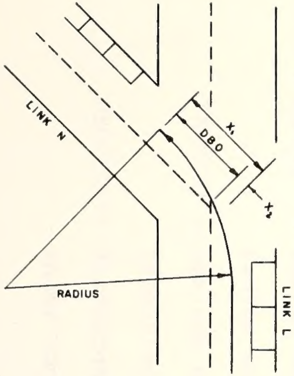
WIDTH = LANE WIDTH
RADIUS = DELAYED TURN RADIUS
(FOR DBO(6))
RADIUS = FREE TURN RADIUS
(FOR DBO(7))

$$DBO(6,7) = RADIUS - WIDTH/2$$

FIGURE 15d.

FIGURE 15, cont.

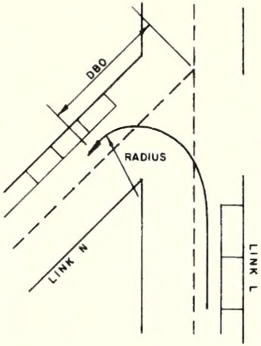
COMPUTATION OF DISCHARGE BOUNDARY OFFSET



WIDTH = LANE WIDTH
W = STREET WIDTH
PARKW = PARKING LANE WIDTH
RADIUS = DELAYED TURN RADIUS
(FOR DBO(4))
RADIUS = FREE TURN RADIUS
(FOR DBO(5))

$$\begin{aligned}\text{DBO}(4,5) &= X_1 - X_2\\ &= \text{RADIUS}(\text{TAN}(22.5^\circ))\\ &\quad -(\text{W(N)}/2-\text{WIDTH}/2-\text{PARKW(N)}/2)\,\text{TAN}(22.5^\circ)\end{aligned}$$

FIGURE 15e.



WIDTH = LANE WIDTH
W = STREET WIDTH
PARKW = PARKING LANE WIDTH
NOLANE = NUMBER OF LANES IN LINK
RADIUS = DELAYED TURN RADIUS
(FOR DBO(8))
RADIUS = FREE TURN RADIUS
(FOR DBO(9))

$$\begin{aligned}\text{DBO}(8,9) = {}&(\text{RADIUS}-\text{W(N)}+\text{WIDTH}+\text{PARKW})2\\ &+(\text{RADIUS}+\text{WIDTH}(\text{NOLANE(L)}-/\,)\\ &-\text{W(L)}/2+\text{PARKW(L)})\sqrt{2}-\text{RADIUS}\\ &+\text{W(N)}-\text{WIDTH}-\text{PARKW(N)}\end{aligned}$$

FIGURE 15f.

FIGURE 15, cont.

COMPUTATION OF DISCHARGE BOUNDARY OFFSET

discharge boundaries for each link. Referring to Figure 16, the beginning of turn point is computed by the general equation

BEGINPT = COI - RADFACT

where

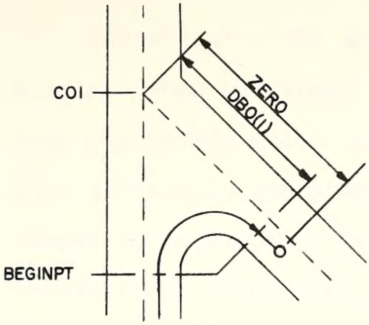
BEGINPT = beginning of turn point;
COI = redefined center of intersection; and
RADFACT = a radius factor, dependent on the radius of the turning movement, link widths, and parking lane width. It is defined in Figure 16 for the various turning movement types.

The accompanying discharge boundary may also be computed, using the general equation

DBNDRY = BEGINPT + LTURN + (ZERO(N) - DBO(NT))

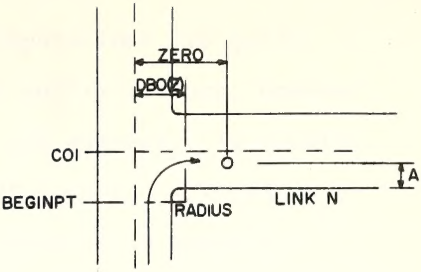
where

DBNDRY = discharge boundary for turning movement;
LTURN = length of turn, calculated from radius and angle of turn;
ZERO(N) = maximum discharge boundary offset of receiving link N; and
DBO(NT) = discharge boundary offset of turn under consideration, NT.



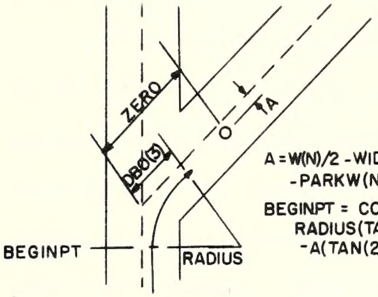
BEGINPT = COI - DBO(1)
DBNDRY = BEGINPT + 3/4 π(RADIUS) + (ZERO - DBO(1))

(a)



A = PARKW(N) + WIDTH/2
BEGINPT = COI - RADIUS - (W(N)/2-A)
DBNDRY = BEGINPT + π/2 (RADIUS) + (ZERO - DBO(2))

(b)

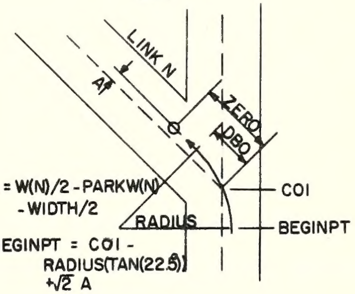


A = W(N)/2 - WIDTH/2 - PARKW(N)

BEGINPT = COI - RADIUS(TAN(22.5°)) - A(TAN(22.5°))

DBNDRY = BEGINPT + π/4 (RADIUS) + (ZERO - DBO(3))

(c)

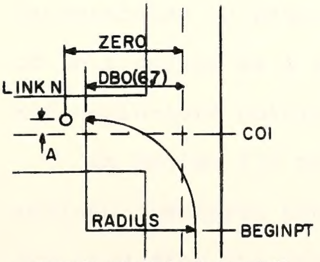


A = W(N)/2 - PARKW(N) - WIDTH/2

BEGINPT = COI - RADIUS(TAN(22.5°)) + $\sqrt{2}$ A

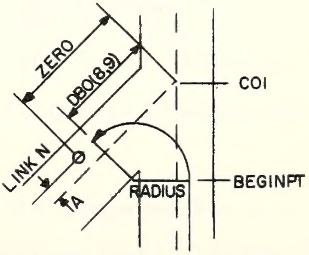
DBNDRY = BEGINPT + π/4 (RADIUS) + (ZERO - DBO(4,5))

(d)



A = (W(N)/2 - WIDTH/2 - PARKW(N))
BEGINPT = COI - RADIUS + A
DBNDRY = BEGINPT + 3π/4 (RADIUS) + (ZERO - DBO (6,7))

(e)



A = W(N)/2 - WIDTH/2 - PARKW(N)
BEGINPT = COI - DBO(8,9) + $\sqrt{2}$ A
DBNDRY = BEGINPT + 3π/4 (RADIUS) + (ZERO - DBO(8,9))

(f)

FIGURE 16
DISCHARGE BOUNDARY COMPUTATIONS

Individually the above computations are quite simple. When performed for an entire network, however, they can become quite tedious and awkward. To resolve this difficulty the program PRESIG was developed as a companion to SIGNET. Its function is to compute a complete set of geometry data ready for input to SIGNET, based on easily acquired measurements from the network. This is accomplished by following the general computational procedure outlined above.

For purposes of discussion the logic of PRESIG may be divided into 3 areas as shown in Figure 17. Region I contains the primary input and initialization functions. Included in the input block is the verification of the input data, primarily achieved through checking card types and link numbers. Additional means for verification are obtained by printing out all input data.

Region II is concerned with the computations demonstrated on pages 57 through 63. The final product of this series is a tentative set of discharge boundaries and begin-turn points for each link.

In Region III this tentative set is compared to revised discharge boundaries and begin-turn points which are read from the input data. Any differences are changed to agree with the revised values. This option is provided to enable the user to easily correct any discrepancies which may arise between the PRESIG output

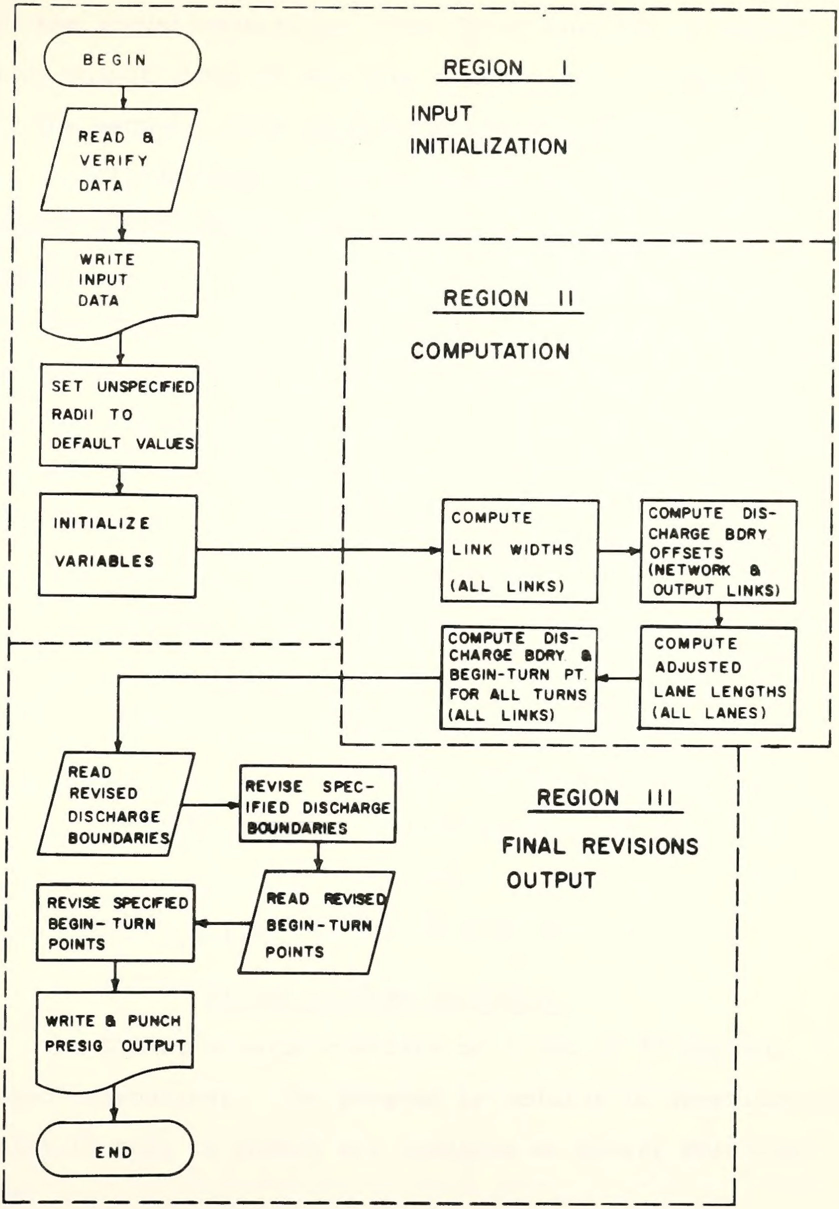

BEGIN
READ & VERIFY DATA
WRITE INPUT DATA
SET UNSPECIFIED RADII TO DEFAULT VALUES
INITIALIZE VARIABLES
REGION I
INPUT INITIALIZATION
REGION II
COMPUTATION
COMPUTE LINK WIDTHS (ALL LINKS)
COMPUTE DISCHARGE BDRY OFFSETS (NETWORK & OUTPUT LINKS)
COMPUTE ADJUSTED LANE LENGTHS (ALL LANES)
COMPUTE DISCHARGE BDRY & BEGIN-TURN PT FOR ALL TURNS (ALL LINKS)
READ REVISED DISCHARGE BOUNDARIES
REVISE SPECIFIED DISCHARGE BOUNDARIES
READ REVISED BEGIN-TURN POINTS
REVISE SPECIFIED BEGIN-TURN POINTS
WRITE & PUNCH PRESIG OUTPUT
END
REGION III
FINAL REVISIONS OUTPUT

FIGURE 17

PRESIG FLOWCHART

and the actual conditions. The final function of PRESIG is to output, both on the line printer and card punch, all the geometry data needed for input to SIGNET.

A full description of the inputs to PRESIG and a listing of the PRESIG program code is provided in Appendix A.

The output from PRESIG is in two forms; printed and punched. Each form includes all the geometry data required for input to SIGNET; the punched output therefore may be used directly as a portion of the SIGNET input deck.

The following variables are output for each link:

STPLINE = coordinate of the stop line;

$XLINK_{j,j=1,3}$ = length of each lane, j;

$DBNDRY_{j,j=1,6}$ = discharge boundary for each turning movement, j;

$BEGINPT_{j,j=1,6}$ = beginning point of each turn, j; and

$RAD_{j,j=1,6}$ = radius of each turn, j.

SIGNET Program Structure

The SIGNET program consists of a set of 33 nested, closed subroutines. The program is modular in construction, making it easy to insert new routines or modify existing ones.

The structure of the SIGNET program is shown in Figure 18. It is seen that 28 routines occupy a fixed place in the program structure, while 5 routines serve auxiliary functions which are called upon at various places in the program. All routines are written in FORTRAN IV for operation under the MACE operating system on the CDC 6500 computer. A complete listing of the SIGNET program code and a full description of all inputs are given in Appendix B.

Tasks performed by the main program SIGNET include:

1. Input of all program data, program instructions, and link data, followed by formatting and writing of these data.

2. Primary initializing tasks, which include initializing all variables not connected with the statistical summary.

3. Activation of the traffic data input and initialization routines (INITRN and INIVOL) at the proper times as expressed on the program instruction cards.

4. Maintenance of the time loop, which includes control over the simulation time cycle and proper calling of the vehicle generation, vehicle update, and interlink transfer routines.

5. Program termination, which entails calling the final summary routine and writing final counts.

The Main Program Chain is shown in Figure 19.

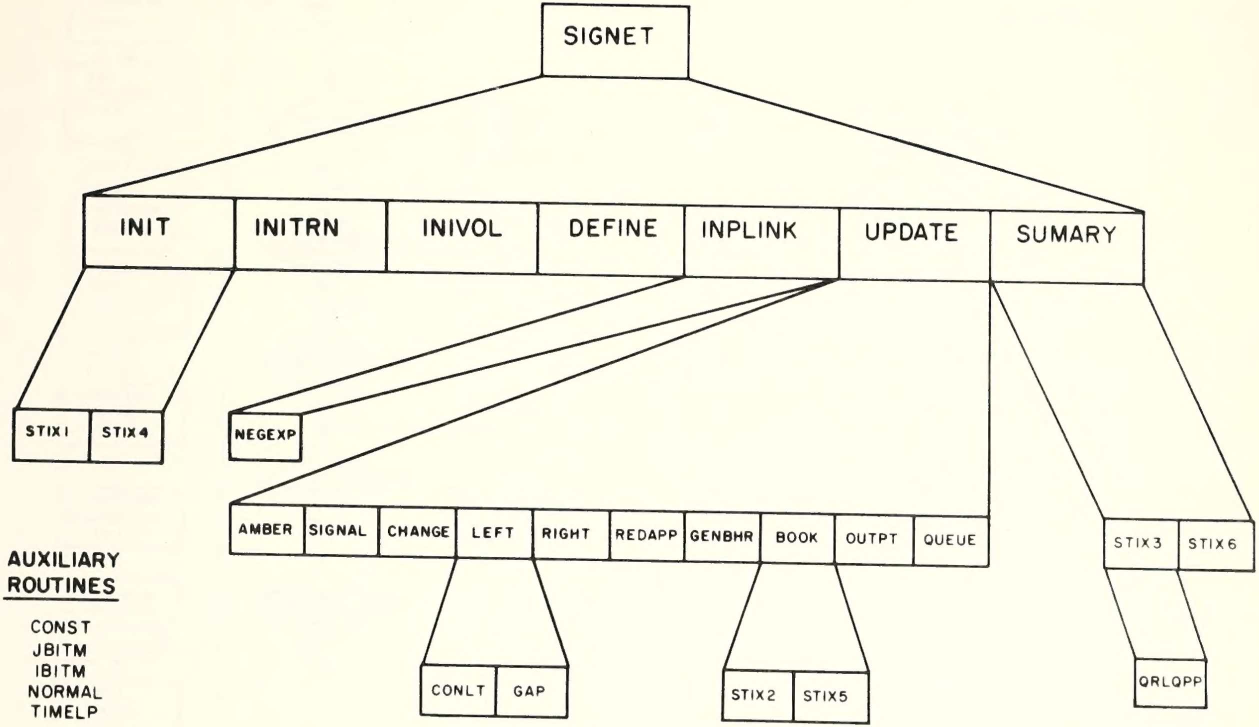


FIGURE 18

SIGNET PROGRAM STRUCTURE

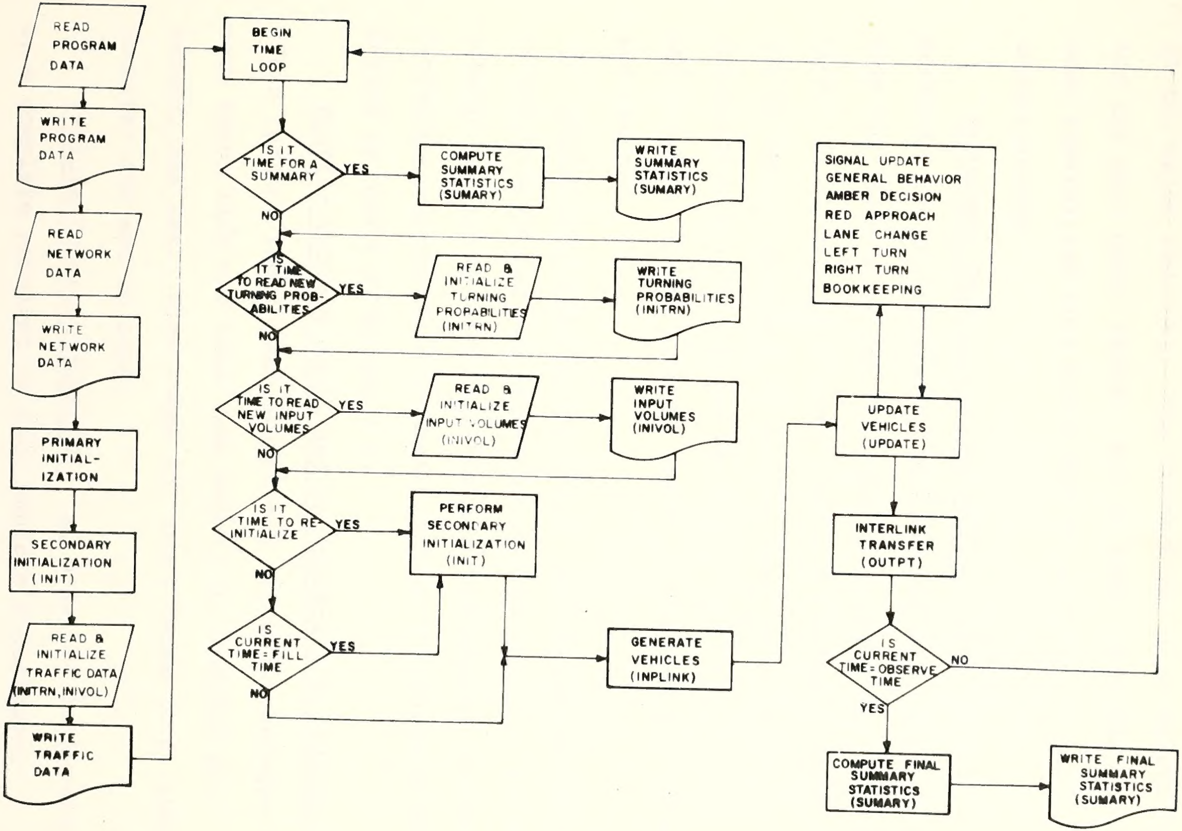

READ PROGRAM DATA
WRITE PROGRAM DATA
READ NETWORK DATA
WRITE NETWORK DATA
PRIMARY INITIAL-IZATION
SECONDARY INITIALIZATION (INIT)
READ & INITIALIZE TRAFFIC DATA (INITRN, INIVOL)
WRITE TRAFFIC DATA
BEGIN TIME LOOP
IS IT TIME FOR A SUMMARY
YES
NO
COMPUTE SUMMARY STATISTICS (SUMARY)
WRITE SUMMARY STATISTICS (SUMARY)
IS IT TIME TO READ NEW TURNING PROB-ABILITIES
YES
NO
READ & INITIALIZE TURNING PROBABILITIES (INITRN)
WRITE TURNING PROBABILITIES (INITRN)
IS IT TIME TO READ NEW INPUT VOLUMES
YES
NO
READ & INITIALIZE INPUT VOLUMES (INIVOL)
WRITE INPUT VOLUMES (INIVOL)
IS IT TIME TO RE-INITIALIZE
YES
NO
PERFORM SECONDARY INITIALIZATION (INIT)
IS CURRENT TIME = FILL TIME
YES
NO
GENERATE VEHICLES (INPLINK)
SIGNAL UPDATE
GENERAL BEHAVIOR
AMBER DECISION
RED APPROACH
LANE CHANGE
LEFT TURN
RIGHT TURN
BOOKKEEPING
UPDATE VEHICLES (UPDATE)
INTERLINK TRANSFER (OUTPT)
IS CURRENT TIME = OBSERVE TIME
NO
YES
COMPUTE FINAL SUMMARY STATISTICS (SUMARY)
WRITE FINAL SUMMARY STATISTICS (SUMARY)

FIGURE 19
MAIN PROGRAM CHAIN

In the following paragraphs descriptions are given of the major routines. The subroutine flow charts are, for the most part, verbal, and in some instances have been generalized to a minor degree to enhance rapid understanding.

Secondary Initialization (INIT). This routine performs the initialization tasks necessary after fill-time has been completed in order to prepare for observe time. It is also used when reinitialization is called for at other times in the program (see Appendix B). Bookkeeping minimum values are reset to large numbers and counters, sums, sums of squares, and maximum values are cleared.

Turning Probability Input and Initialization (INITRN). In addition to reading the turning probabilities for each network and input link this routine initializes various variables associated with them. A full description of the inputs is given in Appendix B.

Traffic Volumes Input and Initialization (INIVOL). After reading the input volumes and truck percentages for each input link the lane flow rates, in vehicles per second, are set. Flow rates for short lanes (i.e. turning pockets) are set to zero.

Vehicle Printout (DEFINE). As detailed in Appendix B, a complete printout of each vehicle's position, velocity, acceleration, acceleration type, and turning

movement during each cycle may be specified. This routine unpacks the ICAR word, prepares the applicable variables, and writes them on the output file when called by the main program SIGNET.

Vehicle Generation (INPLINK). This routine generates, on a per-lane basis, each vehicle and its associated parameters; the generation time of the next vehicle on the lane is then determined. The flowchart is shown in Figure 20.

Vehicle Update (UPDATE). This routine (see Figure 21) determines the applicability of, and controls the transfer to, the subsidiary behavior routines; namely, lane change, general behavior, amber decision, red approach, right-turn, and left-turn. Additionally it controls the use of the signal update and bookkeeping routines. Based on the acceleration determined in the applicable behavior routine, it calculates the vehicle's velocity and position.

Lane Change (CHANGE). This routine (see Figure 22) determines for each lane if a potential lane changer exists, computes the size of the lane-change lag, and, by a random process, determines its acceptability. As appropriate, the vehicle is moved to the new lane and list modifications performed.

Traffic Signal Update (SIGNAL). This routine (see Figure 23) returns the current signal indication for the

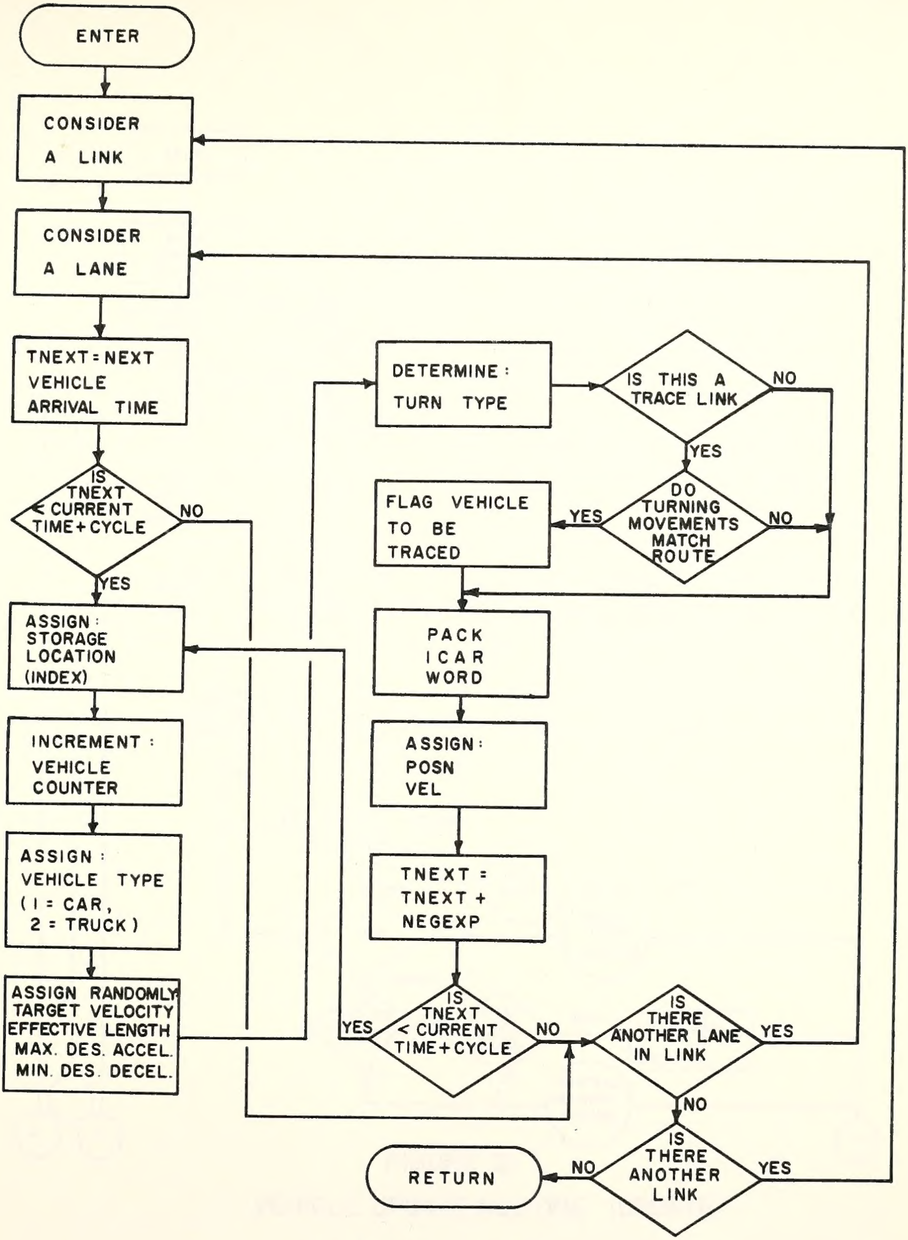

ENTER
CONSIDER A LINK
CONSIDER A LANE
TNEXT = NEXT VEHICLE ARRIVAL TIME
IS TNEXT < CURRENT TIME + CYCLE
NO
YES
ASSIGN : STORAGE LOCATION (INDEX)
INCREMENT : VEHICLE COUNTER
ASSIGN : VEHICLE TYPE (1 = CAR, 2 = TRUCK)
ASSIGN RANDOMLY TARGET VELOCITY EFFECTIVE LENGTH MAX. DES. ACCEL. MIN. DES. DECEL.
DETERMINE : TURN TYPE
IS THIS A TRACE LINK
NO
YES
DO TURNING MOVEMENTS MATCH ROUTE
YES
NO
FLAG VEHICLE TO BE TRACED
PACK ICAR WORD
ASSIGN : POSN VEL
TNEXT = TNEXT + NEGEXP
IS TNEXT < CURRENT TIME + CYCLE
YES
NO
IS THERE ANOTHER LANE IN LINK
YES
NO
IS THERE ANOTHER LINK
YES
NO
RETURN

FIGURE 20

VEHICLE GENERATION ROUTINE (INPLINK)

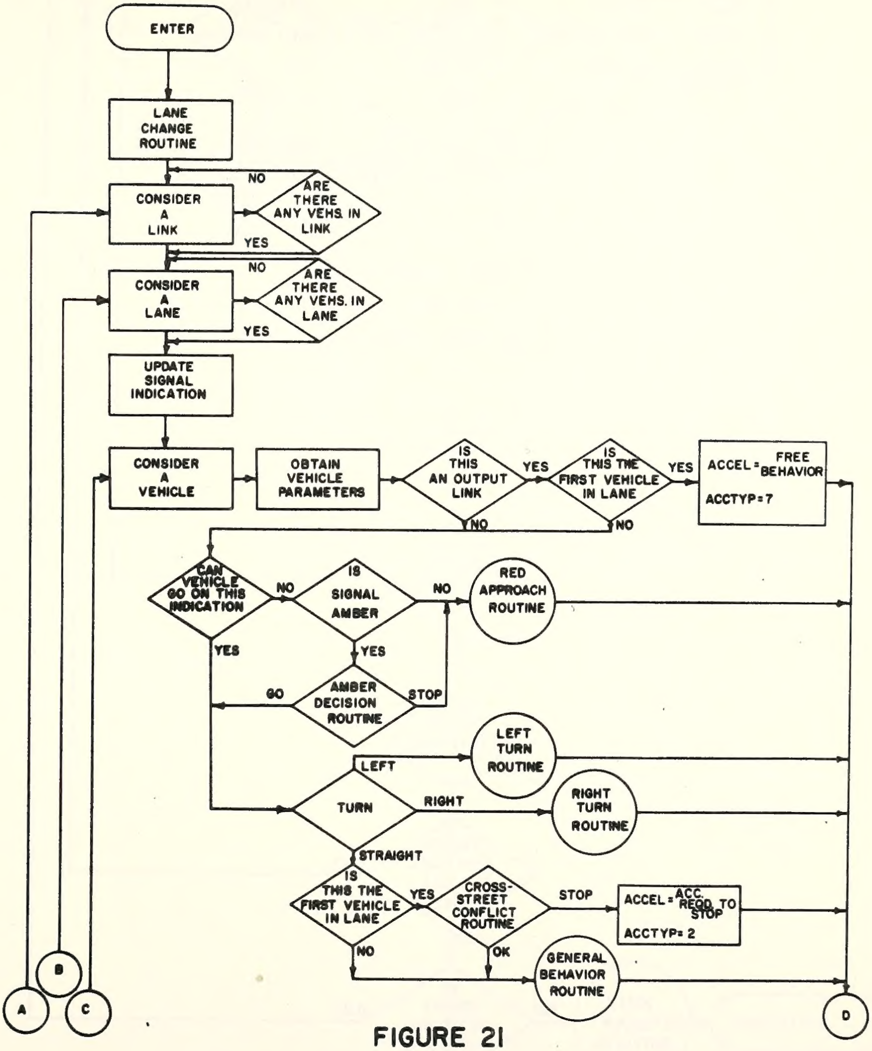


FIGURE 21

VEHICLE UPDATE ROUTINE (UPDATE)

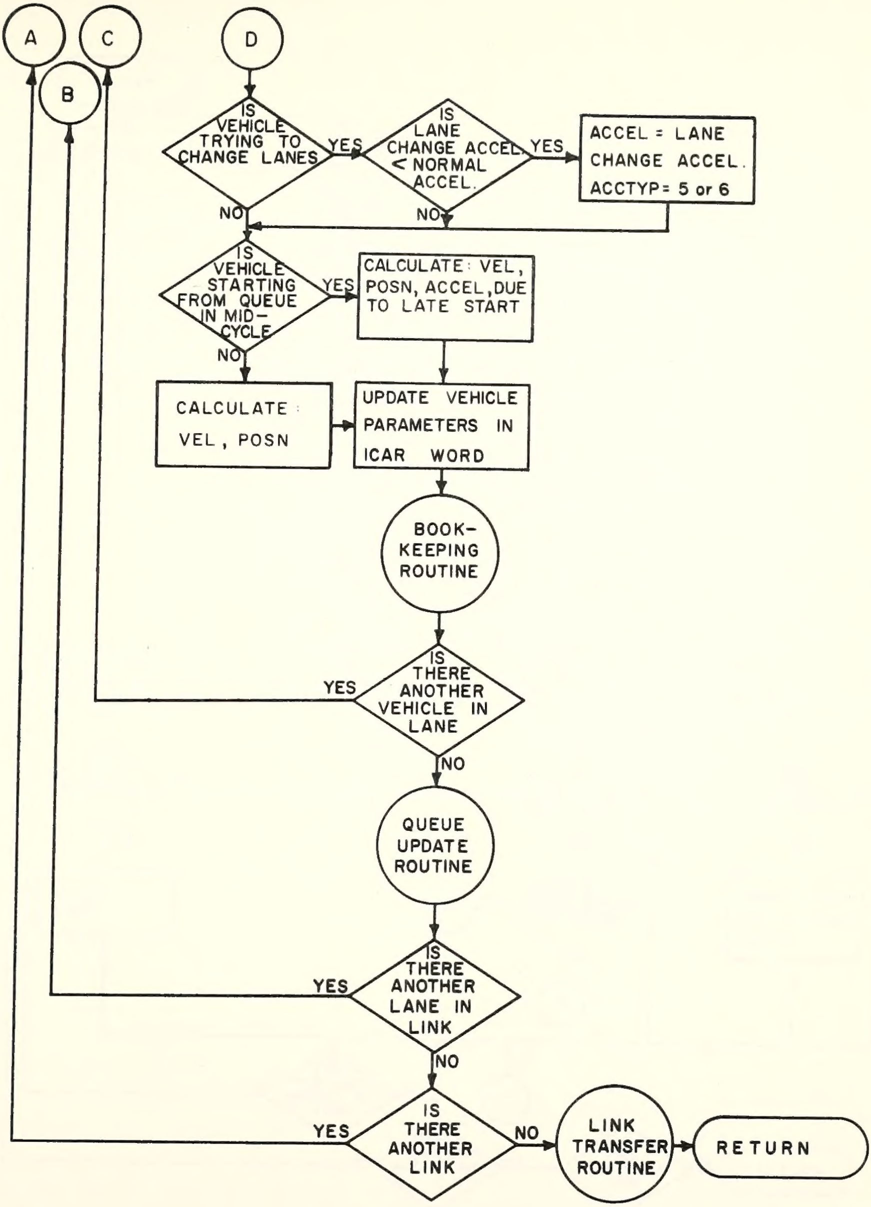

A
B
C
D
IS VEHICLE TRYING TO CHANGE LANES
YES
NO
IS LANE CHANGE ACCEL < NORMAL ACCEL.
YES
NO
ACCEL = LANE CHANGE ACCEL.
ACCTYP= 5 or 6
IS VEHICLE STARTING FROM QUEUE IN MID-CYCLE
YES
NO
CALCULATE: VEL, POSN, ACCEL, DUE TO LATE START
CALCULATE: VEL, POSN
UPDATE VEHICLE PARAMETERS IN ICAR WORD
BOOK-KEEPING ROUTINE
IS THERE ANOTHER VEHICLE IN LANE
YES
NO
QUEUE UPDATE ROUTINE
IS THERE ANOTHER LANE IN LINK
YES
NO
IS THERE ANOTHER LINK
YES
NO
LINK TRANSFER ROUTINE
RETURN

FIGURE 21, cont.

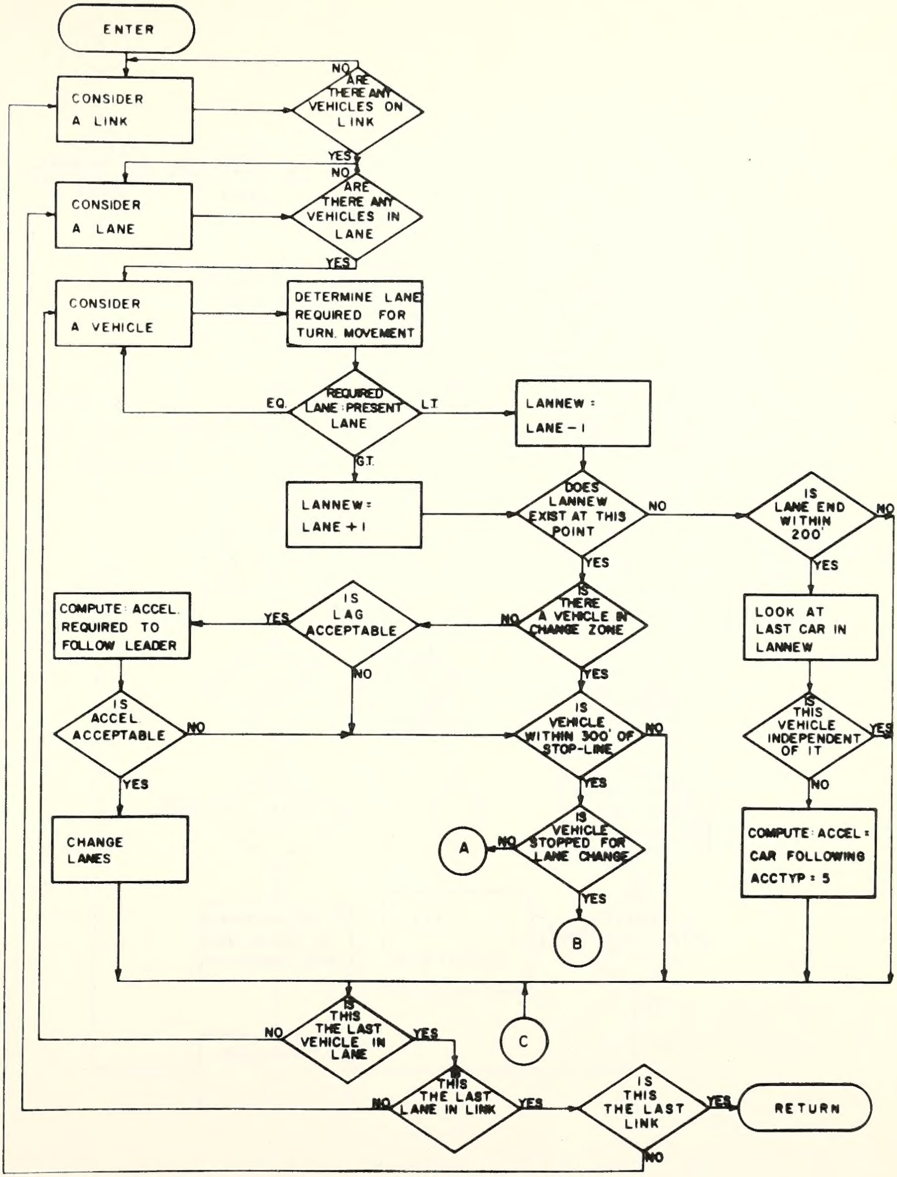

ENTER
CONSIDER A LINK
ARE THERE ANY VEHICLES ON LINK
NO
YES
CONSIDER A LANE
ARE THERE ANY VEHICLES IN LANE
NO
YES
CONSIDER A VEHICLE
DETERMINE LANE REQUIRED FOR TURN. MOVEMENT
REQUIRED LANE : PRESENT LANE
EQ.
L.T.
G.T.
LANNEW = LANE − 1
LANNEW = LANE + 1
DOES LANNEW EXIST AT THIS POINT
NO
YES
IS LANE END WITHIN 200'
NO
YES
LOOK AT LAST CAR IN LANNEW
IS THIS VEHICLE INDEPENDENT OF IT
YES
NO
COMPUTE: ACCEL = CAR FOLLOWING ACCTYP = 5
IS THERE A VEHICLE IN CHANGE ZONE
NO
YES
IS LAG ACCEPTABLE
YES
NO
COMPUTE: ACCEL. REQUIRED TO FOLLOW LEADER
IS ACCEL. ACCEPTABLE
NO
YES
CHANGE LANES
IS VEHICLE WITHIN 300' OF STOP-LINE
NO
YES
IS VEHICLE STOPPED FOR LANE CHANGE
NO
YES
A
B
C
IS THIS THE LAST VEHICLE IN LANE
NO
YES
IS THIS THE LAST LANE IN LINK
NO
YES
IS THIS THE LAST LINK
YES
NO
RETURN

FIGURE 22

LANE CHANGE ROUTINE (CHANGE)

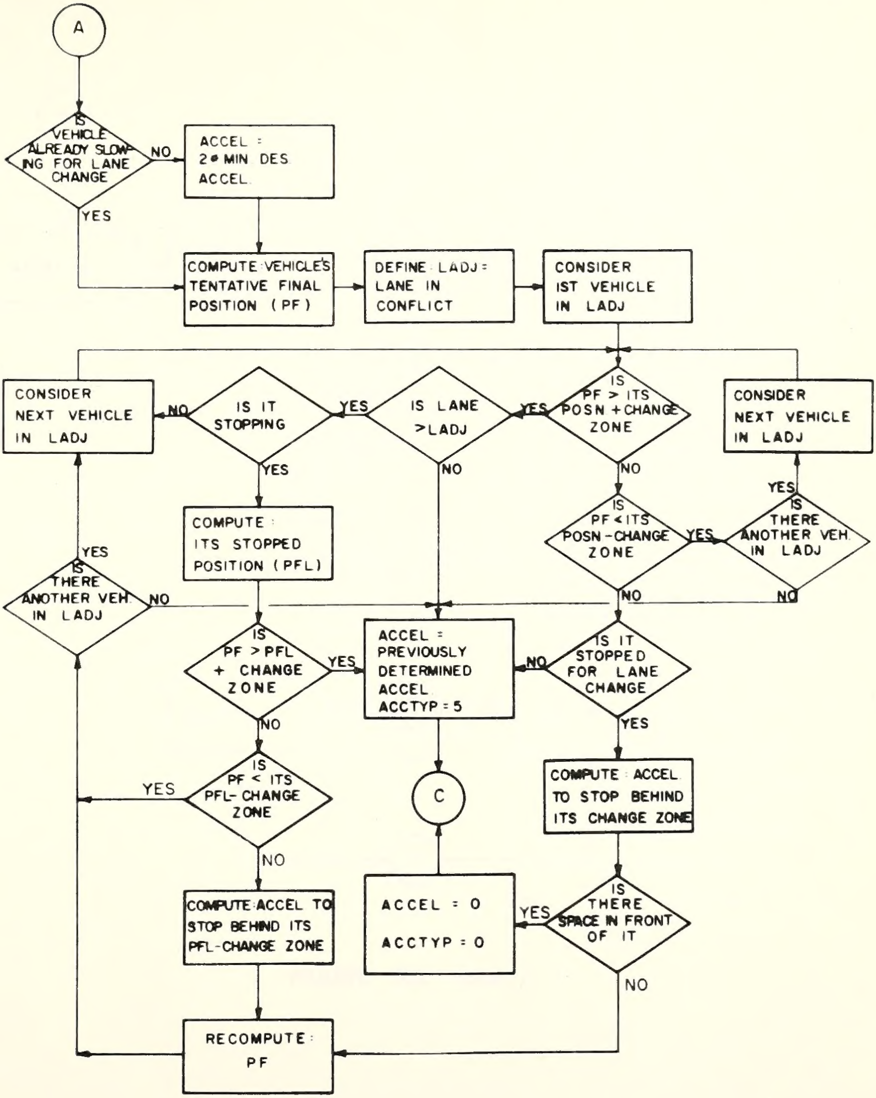

A
IS VEHICLE ALREADY SLOWING FOR LANE CHANGE
NO
YES
ACCEL = 2 * MIN DES ACCEL
COMPUTE VEHICLE'S TENTATIVE FINAL POSITION (PF)
DEFINE LADJ = LANE IN CONFLICT
CONSIDER 1ST VEHICLE IN LADJ
CONSIDER NEXT VEHICLE IN LADJ
NO
IS IT STOPPING
YES
IS LANE > LADJ
YES
IS PF > ITS POSN + CHANGE ZONE
CONSIDER NEXT VEHICLE IN LADJ
YES
NO
NO
COMPUTE ITS STOPPED POSITION (PFL)
IS PF < ITS POSN - CHANGE ZONE
YES
IS THERE ANOTHER VEH IN LADJ
YES
IS THERE ANOTHER VEH IN LADJ
NO
NO
NO
IS PF > PFL + CHANGE ZONE
YES
ACCEL = PREVIOUSLY DETERMINED ACCEL ACCTYP = 5
NO
IS IT STOPPED FOR LANE CHANGE
NO
YES
YES
IS PF < ITS PFL- CHANGE ZONE
C
COMPUTE ACCEL TO STOP BEHIND ITS CHANGE ZONE
NO
COMPUTE ACCEL TO STOP BEHIND ITS PFL-CHANGE ZONE
ACCEL = 0
ACCTYP = 0
YES
IS THERE SPACE IN FRONT OF IT
NO
RECOMPUTE PF

FIGURE 22 (cont.)

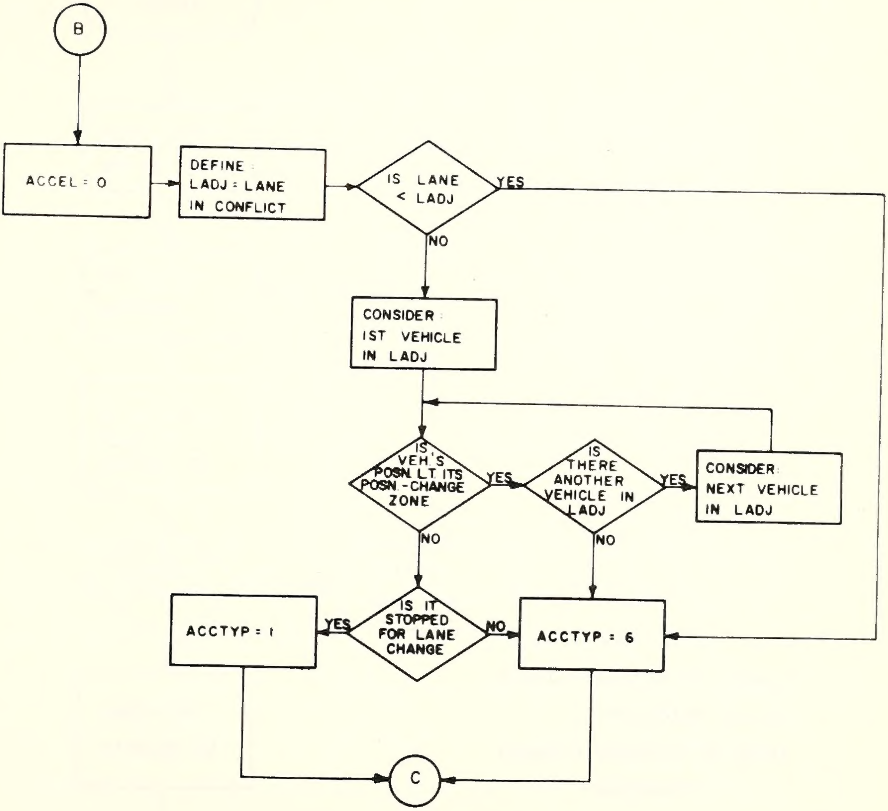
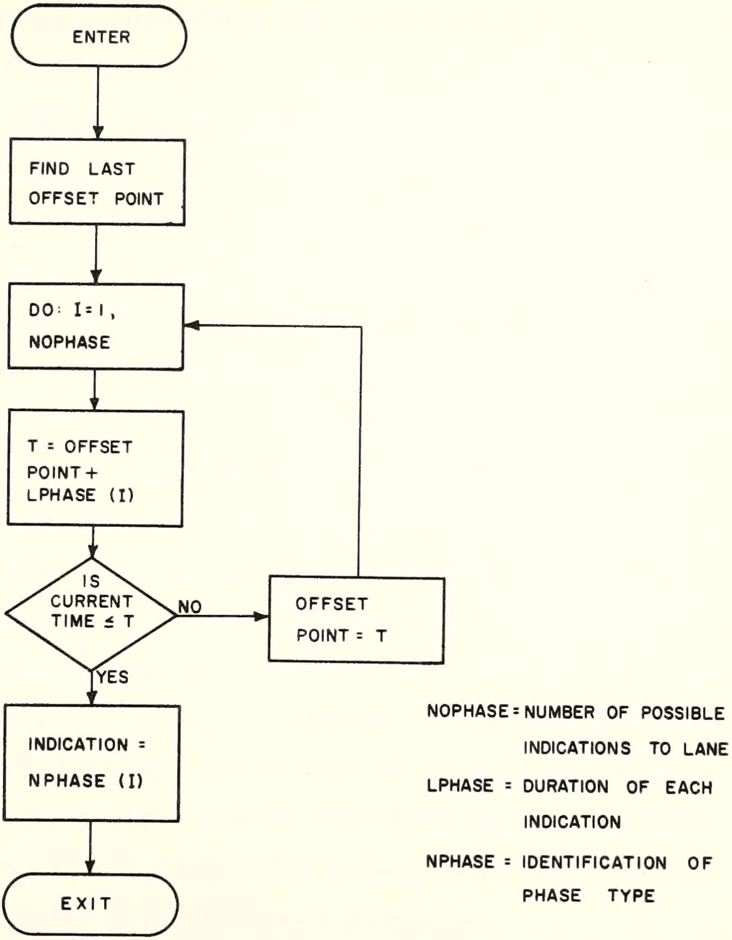


FIGURE 22 (cont.)



NOPHASE = NUMBER OF POSSIBLE INDICATIONS TO LANE
LPHASE = DURATION OF EACH INDICATION
NPHASE = IDENTIFICATION OF PHASE TYPE

FIGURE 23

SIGNAL UPDATE ROUTINE (SIGNAL)

lane under consideration. It is therefore called once for each lane in the network each scan cycle. In its current implementation the routine will operate only in the fixed-time mode.

Amber Decision (AMBER). This routine (see Figure 24) determines if a vehicle approaching the intersection will stop or go through the intersection on amber. If it decides to go, control is transferred either to the general behavior routine or a turning routine, as determined by the vehicle's turning movement. Otherwise control is transferred to the red-approach routine. Decisions are made by a random process related to the stopping acceleration required.

Behavior Routines (GENBHR, REDAPP, RIGHT, LEFT). The four following routines determine the acceleration initiated in the current cycle for each vehicle. Transfer to these routines is controlled by the vehicle-update routine (UPDATE), and is related to both the traffic signal indication and the vehicle's turning movement. A limited amount of transfer occurs among the routines, notably to facilitate queue discharge.

The general-behavior routine GENBHR (see Figure 25) is applied to all vehicles not within the influence of an intersection and to those which are not turning. Contained within it is the logic related to queue discharge.

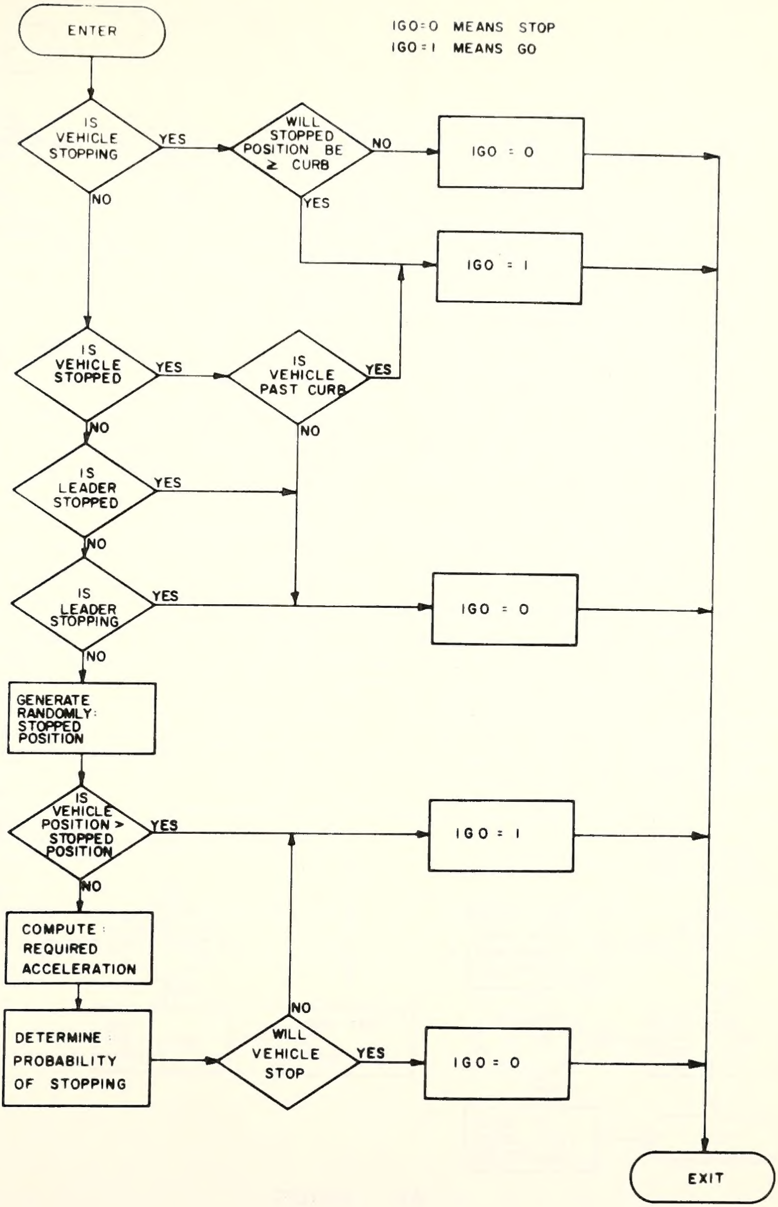


FIGURE 24

AMBER DECISION ROUTINE (AMBER)

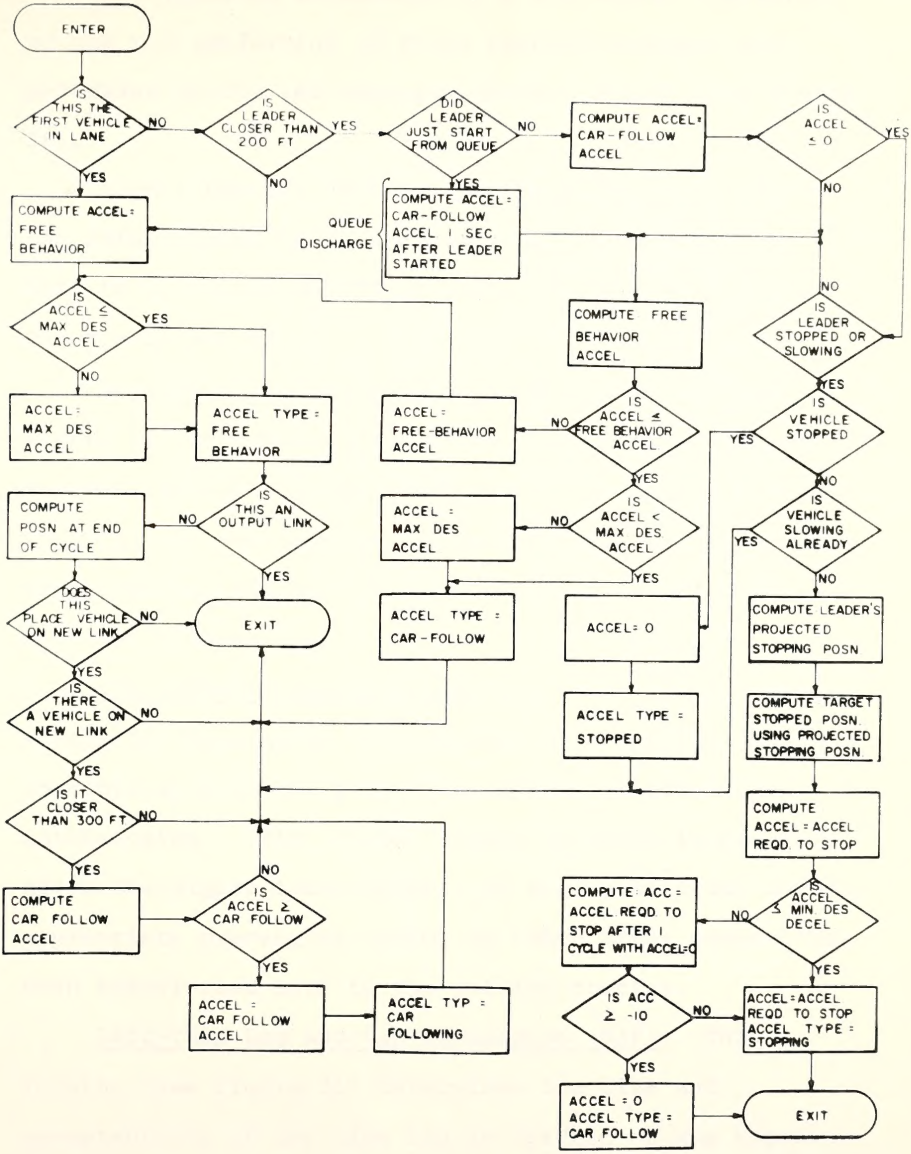

ENTER
IS THIS THE FIRST VEHICLE IN LANE
NO
YES
IS LEADER CLOSER THAN 200 FT
YES
NO
DID LEADER JUST START FROM QUEUE
NO
YES
COMPUTE ACCEL= CAR-FOLLOW ACCEL
IS ACCEL ≤ 0
YES
NO
COMPUTE ACCEL= FREE BEHAVIOR
QUEUE DISCHARGE
COMPUTE ACCEL= CAR-FOLLOW ACCEL 1 SEC AFTER LEADER STARTED
IS ACCEL ≤ MAX DES ACCEL
YES
NO
COMPUTE FREE BEHAVIOR ACCEL
IS LEADER STOPPED OR SLOWING
NO
YES
ACCEL= MAX DES ACCEL
ACCEL TYPE = FREE BEHAVIOR
ACCEL = FREE-BEHAVIOR ACCEL
IS ACCEL ≤ FREE BEHAVIOR ACCEL
NO
YES
IS VEHICLE STOPPED
YES
NO
COMPUTE POSN AT END OF CYCLE
IS THIS AN OUTPUT LINK
NO
YES
ACCEL = MAX DES ACCEL
IS ACCEL < MAX DES ACCEL
NO
YES
IS VEHICLE SLOWING ALREADY
YES
NO
DOES THIS PLACE VEHICLE ON NEW LINK
NO
YES
EXIT
ACCEL TYPE = CAR-FOLLOW
ACCEL= 0
COMPUTE LEADER'S PROJECTED STOPPING POSN
IS THERE A VEHICLE ON NEW LINK
NO
YES
ACCEL TYPE = STOPPED
COMPUTE TARGET STOPPED POSN USING PROJECTED STOPPING POSN
IS IT CLOSER THAN 300 FT
NO
YES
COMPUTE ACCEL = ACCEL REQD TO STOP
NO
IS ACCEL ≥ CAR FOLLOW
YES
COMPUTE CAR FOLLOW ACCEL
COMPUTE ACC = ACCEL REQD TO STOP AFTER 1 CYCLE WITH ACCEL=0
IS ACCEL ≤ MIN DES DECEL
NO
YES
ACCEL = CAR FOLLOW ACCEL
ACCEL TYP = CAR FOLLOWING
IS ACC ≥ -10
NO
YES
ACCEL=ACCEL REQD TO STOP ACCEL TYPE = STOPPING
ACCEL = 0 ACCEL TYPE = CAR FOLLOW
EXIT

FIGURE 25

GENERAL BEHAVIOR ROUTINE (GENBHR)

All vehicles confronted by a red signal indication or one not conforming to their turning movement are processed by the red-approach routine REDAPP (see Figure 26).

When a turning vehicle is within the influence of the intersection it is processed by either the right or left turn routine (RIGHT or LEFT), respectively. See Figures 27 and 28.

Left-Turn Conflict (CONLT). This routine (see Figure 29) determines if a vehicle on an opposing link is in conflict with the desired path of a left-turning vehicle. If yes, appropriate preventive action is taken after control has been transferred back to the left-turn behavior routine.

Cross-Street Conflict (CONST). This routine (see Figure 30) determines if a vehicle on a crossing link is in conflict with the path of a vehicle entering the intersection. This is most likely to occur immediately after the signal turns green. If such a conflict exists, appropriate preventive action is taken after control has been transferred back to the calling routine.

Left-Turn Lag and Gap Acceptance (GAP). This routine (see Figure 31) determines the size and acceptability of the time lag in traffic on the opposing approach for a left-turn vehcile. If the left-turning vehicle has rejected the lag, the size and acceptability

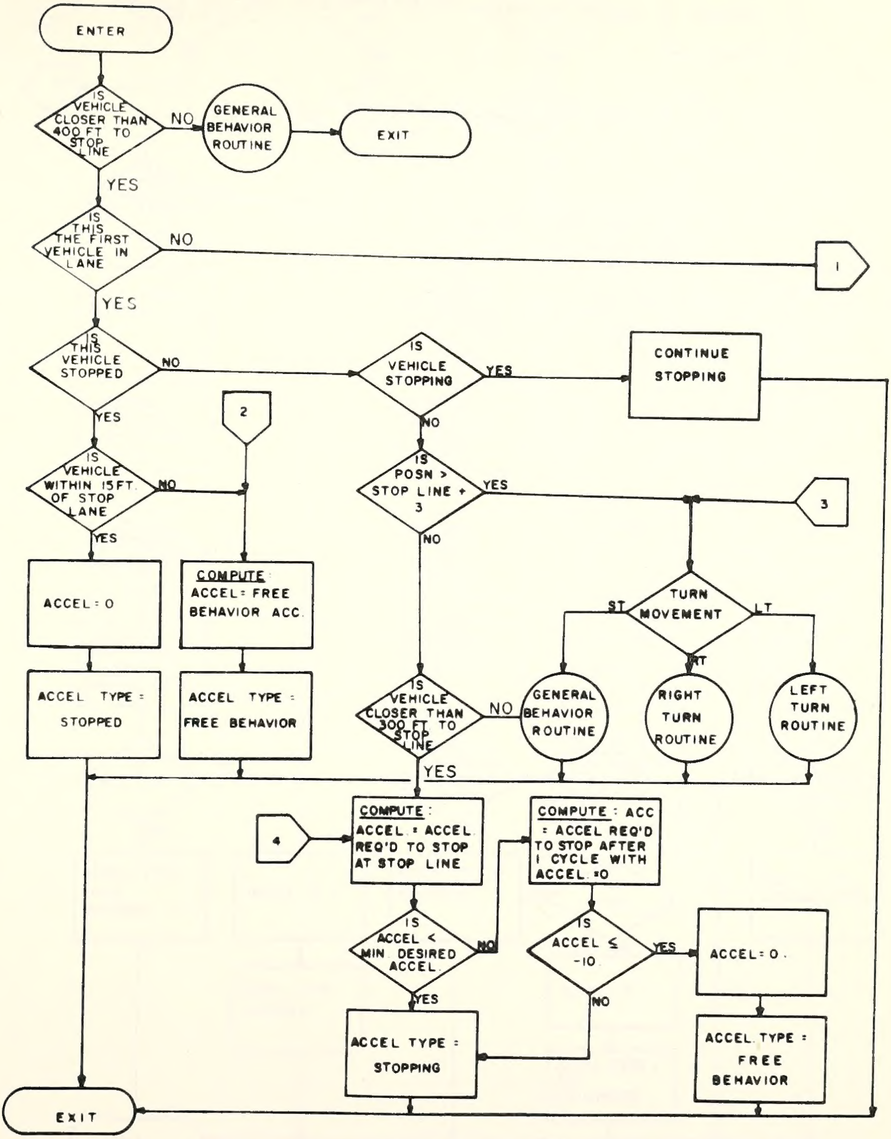

ENTER
IS VEHICLE CLOSER THAN 400 FT TO STOP LINE
NO
GENERAL BEHAVIOR ROUTINE
EXIT
YES
IS THIS THE FIRST VEHICLE IN LANE
NO
1
YES
IS THIS VEHICLE STOPPED
NO
IS VEHICLE STOPPING
YES
CONTINUE STOPPING
YES
2
NO
IS VEHICLE WITHIN 15 FT. OF STOP LANE
NO
IS POSN > STOP LINE + 3
YES
3
YES
NO
ACCEL = 0
COMPUTE: ACCEL = FREE BEHAVIOR ACC.
TURN MOVEMENT
ST
LT
RT
ACCEL TYPE = STOPPED
ACCEL TYPE = FREE BEHAVIOR
IS VEHICLE CLOSER THAN 300 FT TO STOP LINE
NO
GENERAL BEHAVIOR ROUTINE
RIGHT TURN ROUTINE
LEFT TURN ROUTINE
YES
4
COMPUTE: ACCEL = ACCEL REQ'D TO STOP AT STOP LINE
COMPUTE: ACC = ACCEL REQ'D TO STOP AFTER 1 CYCLE WITH ACCEL = 0
IS ACCEL < MIN. DESIRED ACCEL.
NO
IS ACCEL ≤ -10
YES
ACCEL = 0
YES
NO
ACCEL TYPE = STOPPING
ACCEL TYPE = FREE BEHAVIOR
EXIT

FIGURE 26

RED APPROACH ROUTINE (REDAPP)

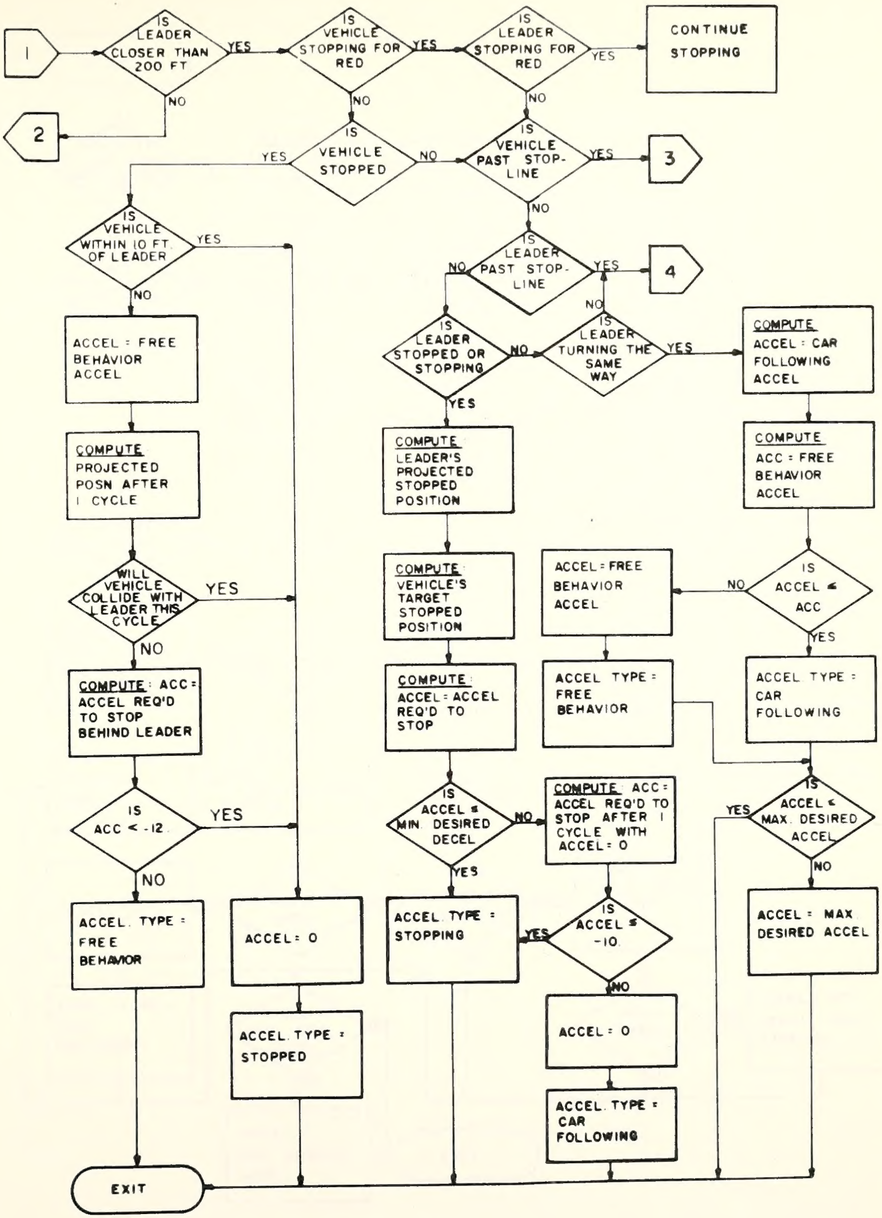

1
IS LEADER CLOSER THAN 200 FT
YES
NO
2
IS VEHICLE STOPPING FOR RED
YES
NO
IS LEADER STOPPING FOR RED
YES
NO
CONTINUE STOPPING
IS VEHICLE STOPPED
YES
NO
IS VEHICLE PAST STOP-LINE
YES
NO
3
IS VEHICLE WITHIN 10 FT. OF LEADER
YES
NO
IS LEADER PAST STOP-LINE
NO
YES
4
NO
ACCEL = FREE BEHAVIOR ACCEL
IS LEADER STOPPED OR STOPPING
NO
IS LEADER TURNING THE SAME WAY
YES
COMPUTE ACCEL = CAR FOLLOWING ACCEL
YES
COMPUTE PROJECTED POSN AFTER 1 CYCLE
COMPUTE LEADER'S PROJECTED STOPPED POSITION
COMPUTE ACC = FREE BEHAVIOR ACCEL
WILL VEHICLE COLLIDE WITH LEADER THIS CYCLE
YES
NO
COMPUTE VEHICLE'S TARGET STOPPED POSITION
ACCEL = FREE BEHAVIOR ACCEL
NO
IS ACCEL ≤ ACC
YES
COMPUTE: ACC = ACCEL REQ'D TO STOP BEHIND LEADER
COMPUTE: ACCEL = ACCEL REQ'D TO STOP
ACCEL. TYPE = FREE BEHAVIOR
ACCEL. TYPE = CAR FOLLOWING
IS ACC < -12
YES
NO
IS ACCEL ≤ MIN. DESIRED DECEL
NO
YES
COMPUTE: ACC = ACCEL REQ'D TO STOP AFTER 1 CYCLE WITH ACCEL = 0
YES
IS ACCEL ≤ MAX. DESIRED ACCEL
NO
ACCEL. TYPE = FREE BEHAVIOR
ACCEL = 0
ACCEL. TYPE = STOPPING
YES
IS ACCEL ≤ -10.
NO
ACCEL = MAX. DESIRED ACCEL
ACCEL. TYPE = STOPPED
ACCEL = 0
ACCEL. TYPE = CAR FOLLOWING
EXIT

FIGURE 26, cont.

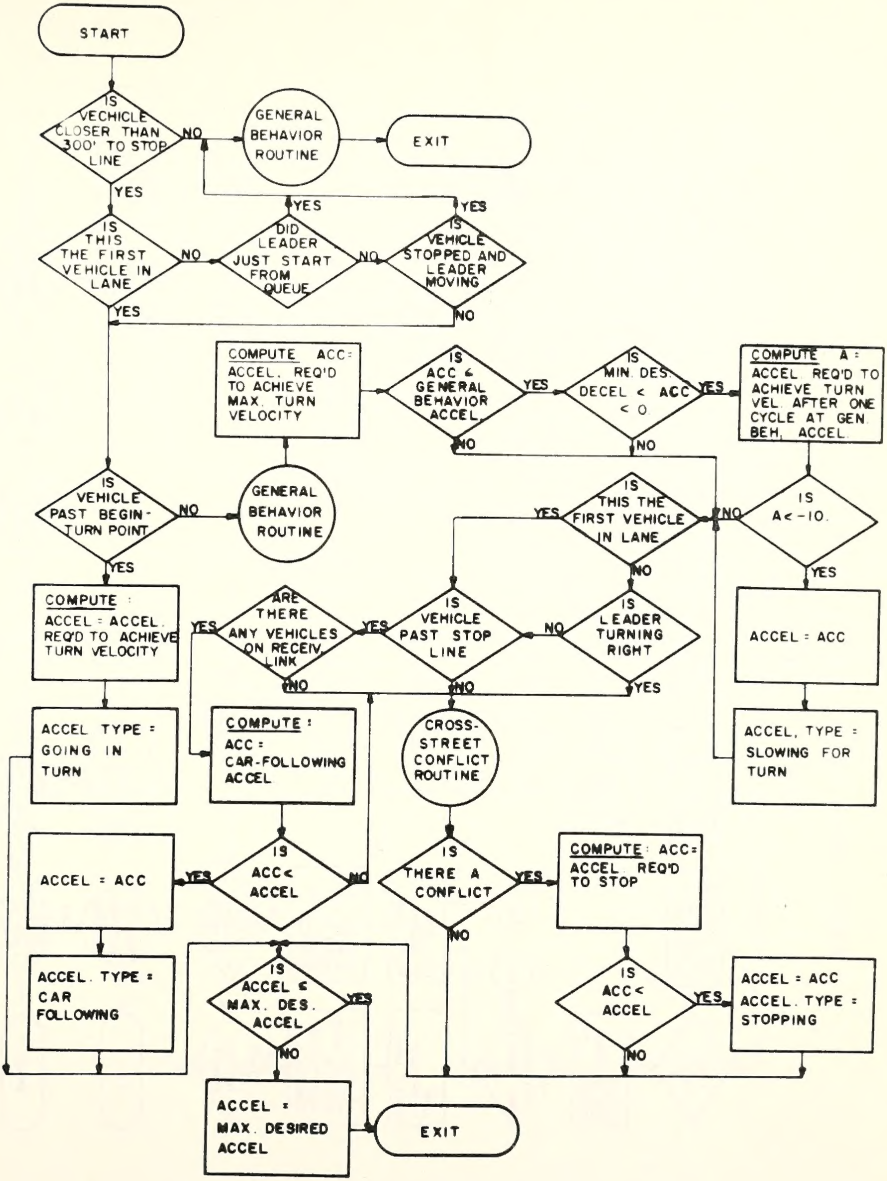

START
IS VECHICLE CLOSER THAN 300' TO STOP LINE
NO
YES
GENERAL BEHAVIOR ROUTINE
EXIT
IS THIS THE FIRST VEHICLE IN LANE
DID LEADER JUST START FROM QUEUE
IS VEHICLE STOPPED AND LEADER MOVING
COMPUTE ACC= ACCEL. REQ'D TO ACHIEVE MAX. TURN VELOCITY
IS ACC ≤ GENERAL BEHAVIOR ACCEL.
IS MIN. DES. DECEL < ACC < 0.
COMPUTE A : ACCEL. REQ'D TO ACHIEVE TURN VEL. AFTER ONE CYCLE AT GEN. BEH. ACCEL.
IS VEHICLE PAST BEGIN-TURN POINT
GENERAL BEHAVIOR ROUTINE
IS THIS THE FIRST VEHICLE IN LANE
IS A< -10.
COMPUTE : ACCEL = ACCEL. REQ'D TO ACHIEVE TURN VELOCITY
ARE THERE ANY VEHICLES ON RECEIV. LINK
IS VEHICLE PAST STOP LINE
IS LEADER TURNING RIGHT
ACCEL = ACC
ACCEL. TYPE = GOING IN TURN
COMPUTE : ACC = CAR-FOLLOWING ACCEL
CROSS-STREET CONFLICT ROUTINE
ACCEL. TYPE = SLOWING FOR TURN
ACCEL = ACC
IS ACC< ACCEL
IS THERE A CONFLICT
COMPUTE : ACC= ACCEL. REQ'D TO STOP
ACCEL. TYPE = CAR FOLLOWING
IS ACCEL ≤ MAX. DES. ACCEL
IS ACC< ACCEL
ACCEL = ACC
ACCEL. TYPE = STOPPING
ACCEL = MAX. DESIRED ACCEL
EXIT

FIGURE 27

RIGHT TURN ROUTINE (RIGHT)

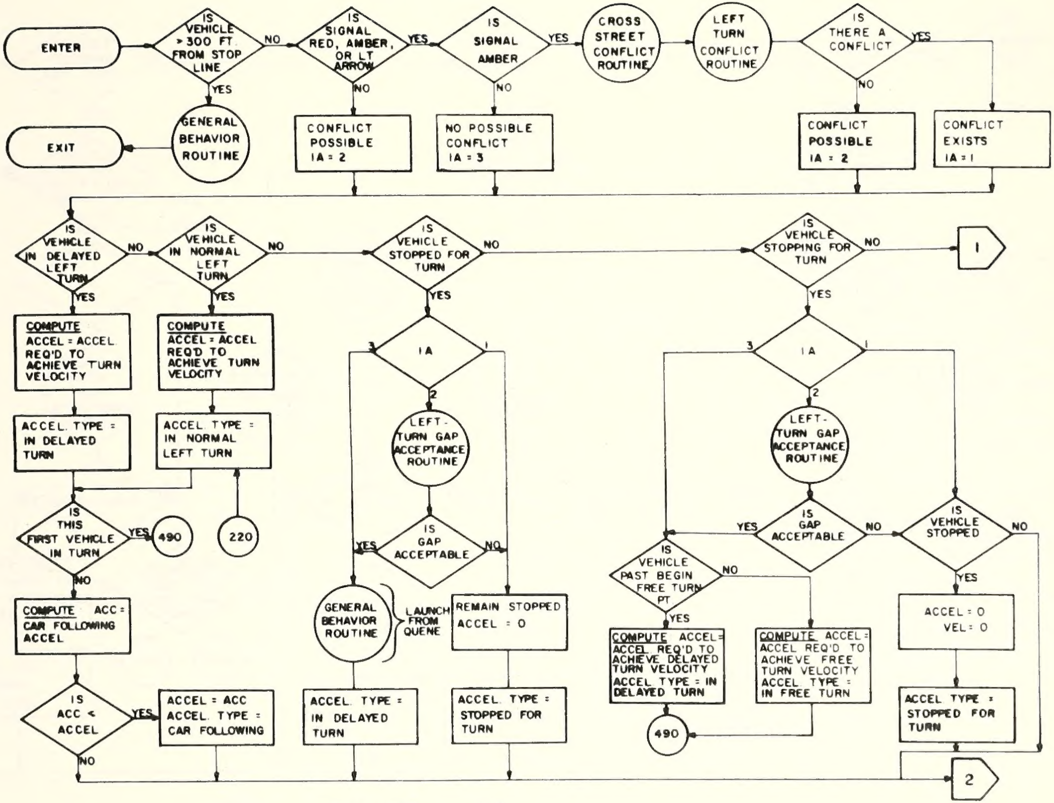

ENTER
IS VEHICLE > 300 FT. FROM STOP LINE
NO
YES
GENERAL BEHAVIOR ROUTINE
EXIT
IS SIGNAL RED, AMBER, OR LT ARROW
YES
NO
CONFLICT POSSIBLE IA = 2
IS SIGNAL AMBER
YES
NO
NO POSSIBLE CONFLICT IA = 3
CROSS STREET CONFLICT ROUTINE
LEFT TURN CONFLICT ROUTINE
IS THERE A CONFLICT
YES
NO
CONFLICT POSSIBLE IA = 2
CONFLICT EXISTS IA = 1
IS VEHICLE IN DELAYED LEFT TURN
NO
YES
COMPUTE ACCEL = ACCEL. REQ'D TO ACHIEVE TURN VELOCITY
ACCEL. TYPE = IN DELAYED TURN
IS VEHICLE IN NORMAL LEFT TURN
NO
YES
COMPUTE ACCEL = ACCEL REQ'D TO ACHIEVE TURN VELOCITY
ACCEL. TYPE = IN NORMAL LEFT TURN
220
IS THIS FIRST VEHICLE IN TURN
YES
490
NO
COMPUTE ACC = CAR FOLLOWING ACCEL
IS ACC < ACCEL
YES
NO
ACCEL = ACC ACCEL. TYPE = CAR FOLLOWING
IS VEHICLE STOPPED FOR TURN
NO
YES
IA
3
2
1
LEFT-TURN GAP ACCEPTANCE ROUTINE
IS GAP ACCEPTABLE
YES
NO
GENERAL BEHAVIOR ROUTINE
LAUNCH FROM QUENE
REMAIN STOPPED ACCEL = 0
ACCEL. TYPE = IN DELAYED TURN
ACCEL. TYPE = STOPPED FOR TURN
IS VEHICLE STOPPING FOR TURN
NO
1
YES
IA
3
2
1
LEFT-TURN GAP ACCEPTANCE ROUTINE
IS GAP ACCEPTABLE
YES
NO
IS VEHICLE PAST BEGIN FREE TURN PT
NO
YES
COMPUTE ACCEL = ACCEL REQ'D TO ACHIEVE DELAYED TURN VELOCITY ACCEL. TYPE = IN DELAYED TURN
490
COMPUTE ACCEL = ACCEL REQ'D TO ACHIEVE FREE TURN VELOCITY ACCEL. TYPE = IN FREE TURN
IS VEHICLE STOPPED
NO
YES
ACCEL = 0 VEL = 0
ACCEL. TYPE = STOPPED FOR TURN
2

FIGURE 28

LEFT TURN ROUTINE (LEFT)

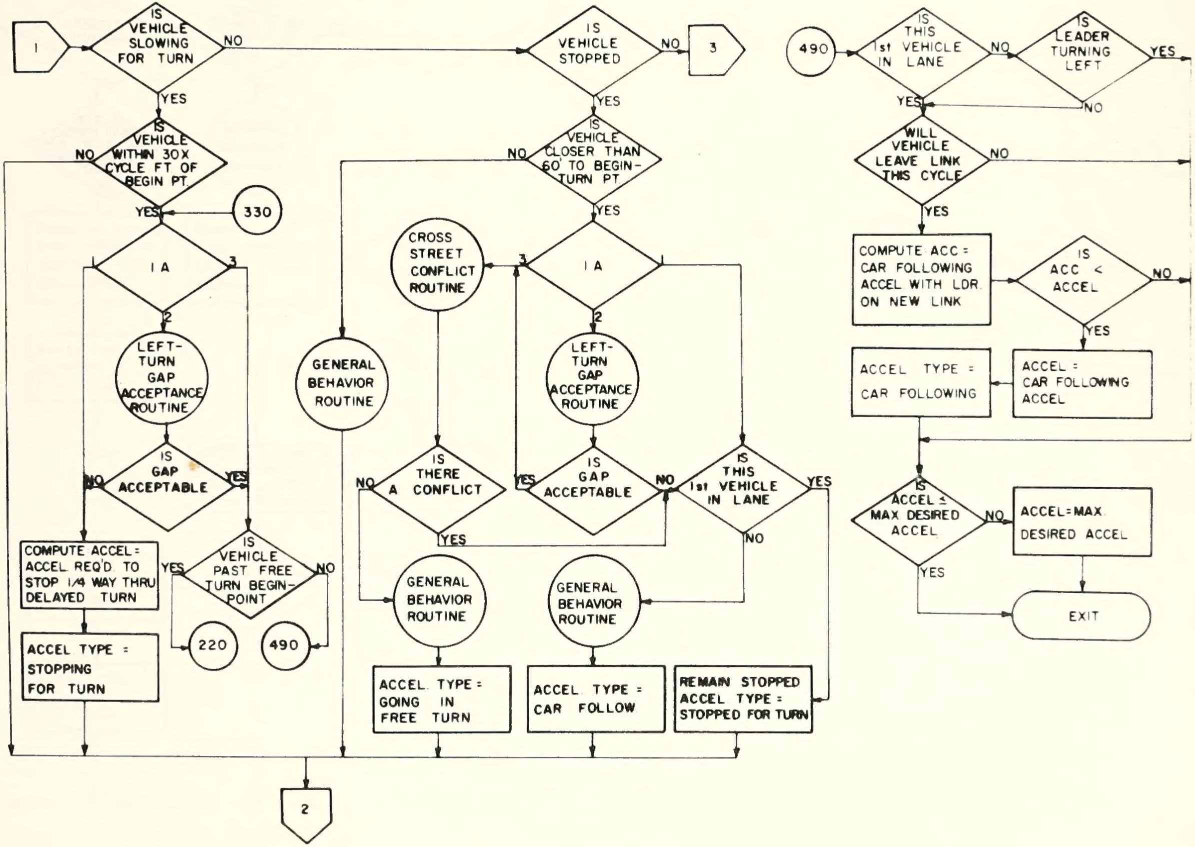

1
IS VEHICLE SLOWING FOR TURN
NO
YES
IS VEHICLE WITHIN 30 X CYCLE FT OF BEGIN PT.
NO
YES
330
I A
1
2
3
LEFT-TURN GAP ACCEPTANCE ROUTINE
IS GAP ACCEPTABLE
NO
YES
COMPUTE ACCEL = ACCEL REQ'D TO STOP 1/4 WAY THRU DELAYED TURN
ACCEL TYPE = STOPPING FOR TURN
IS VEHICLE PAST FREE TURN BEGIN-POINT
YES
NO
220
490
GENERAL BEHAVIOR ROUTINE
IS VEHICLE STOPPED
NO
3
YES
IS VEHICLE CLOSER THAN 60' TO BEGIN-TURN PT.
NO
YES
I A
1
2
3
CROSS STREET CONFLICT ROUTINE
IS THERE A CONFLICT
NO
YES
LEFT-TURN GAP ACCEPTANCE ROUTINE
IS GAP ACCEPTABLE
YES
NO
IS THIS 1st VEHICLE IN LANE
YES
NO
GENERAL BEHAVIOR ROUTINE
ACCEL. TYPE = GOING IN FREE TURN
GENERAL BEHAVIOR ROUTINE
ACCEL. TYPE = CAR FOLLOW
REMAIN STOPPED ACCEL TYPE = STOPPED FOR TURN
2
490
IS THIS 1st VEHICLE IN LANE
NO
YES
IS LEADER TURNING LEFT
YES
NO
WILL VEHICLE LEAVE LINK THIS CYCLE
NO
YES
COMPUTE ACC = CAR FOLLOWING ACCEL WITH LDR. ON NEW LINK
IS ACC < ACCEL
NO
YES
ACCEL = CAR FOLLOWING ACCEL
ACCEL TYPE = CAR FOLLOWING
IS ACCEL ≤ MAX DESIRED ACCEL
NO
YES
ACCEL=MAX. DESIRED ACCEL
EXIT

FIGURE 28, cont.

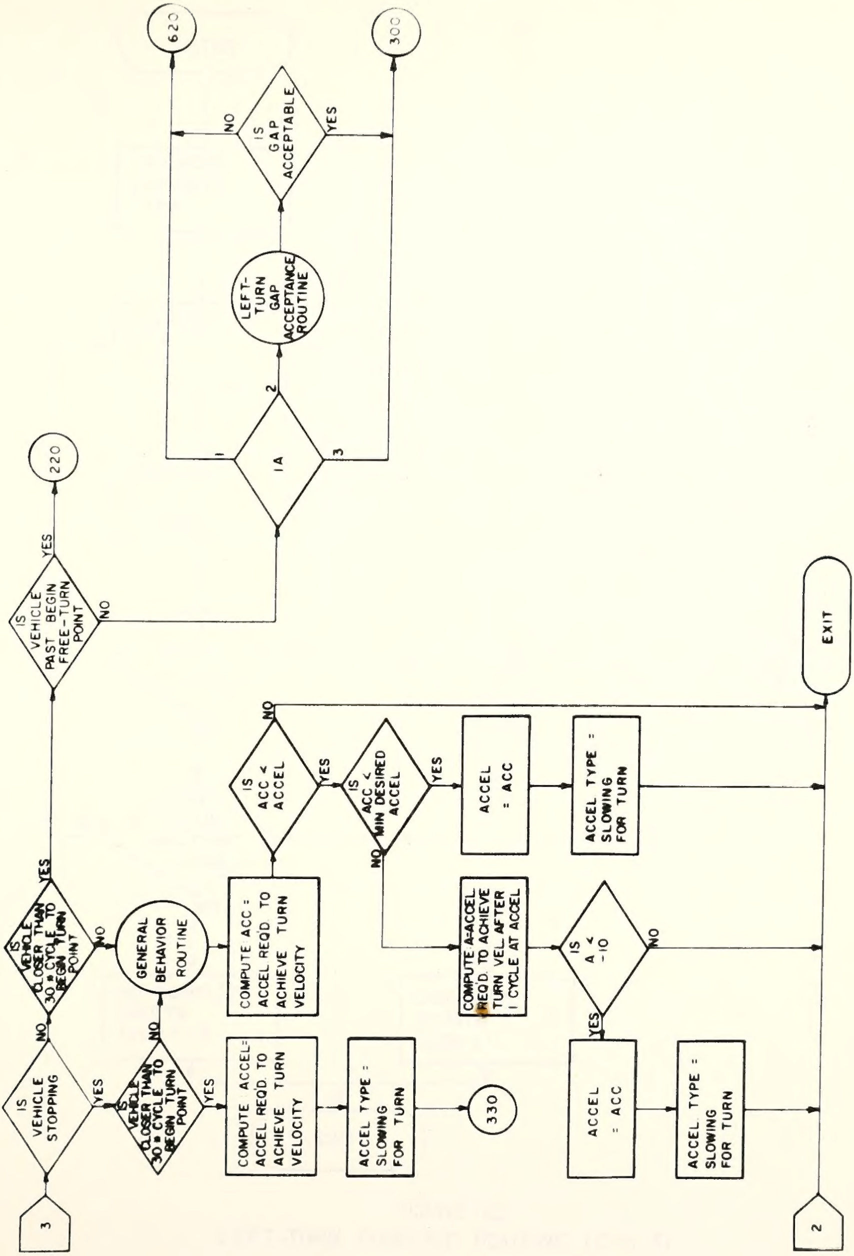

3
IS VEHICLE STOPPING
YES
NO
IS VEHICLE CLOSER THAN 30 * CYCLE TO BEGIN TURN POINT
YES
NO
COMPUTE ACCEL= ACCEL REQ'D. TO ACHIEVE TURN VELOCITY
ACCEL TYPE = SLOWING FOR TURN
330
IS VEHICLE CLOSER THAN 30 * CYCLE TO BEGIN TURN POINT
YES
NO
GENERAL BEHAVIOR ROUTINE
COMPUTE ACC = ACCEL REQ'D. TO ACHIEVE TURN VELOCITY
IS ACC < ACCEL
YES
NO
IS ACC < MIN DESIRED ACCEL
YES
NO
ACCEL = ACC
ACCEL TYPE = SLOWING FOR TURN
COMPUTE A=ACCEL REQ'D. TO ACHIEVE TURN VEL AFTER 1 CYCLE AT ACCEL
IS A < -10
YES
NO
ACCEL = ACC
ACCEL TYPE = SLOWING FOR TURN
2
EXIT
IS VEHICLE PAST BEGIN FREE-TURN POINT
YES
NO
220
IA
1
2
3
LEFT-TURN GAP ACCEPTANCE ROUTINE
IS GAP ACCEPTABLE
NO
YES
620
300

FIGURE 28, cont

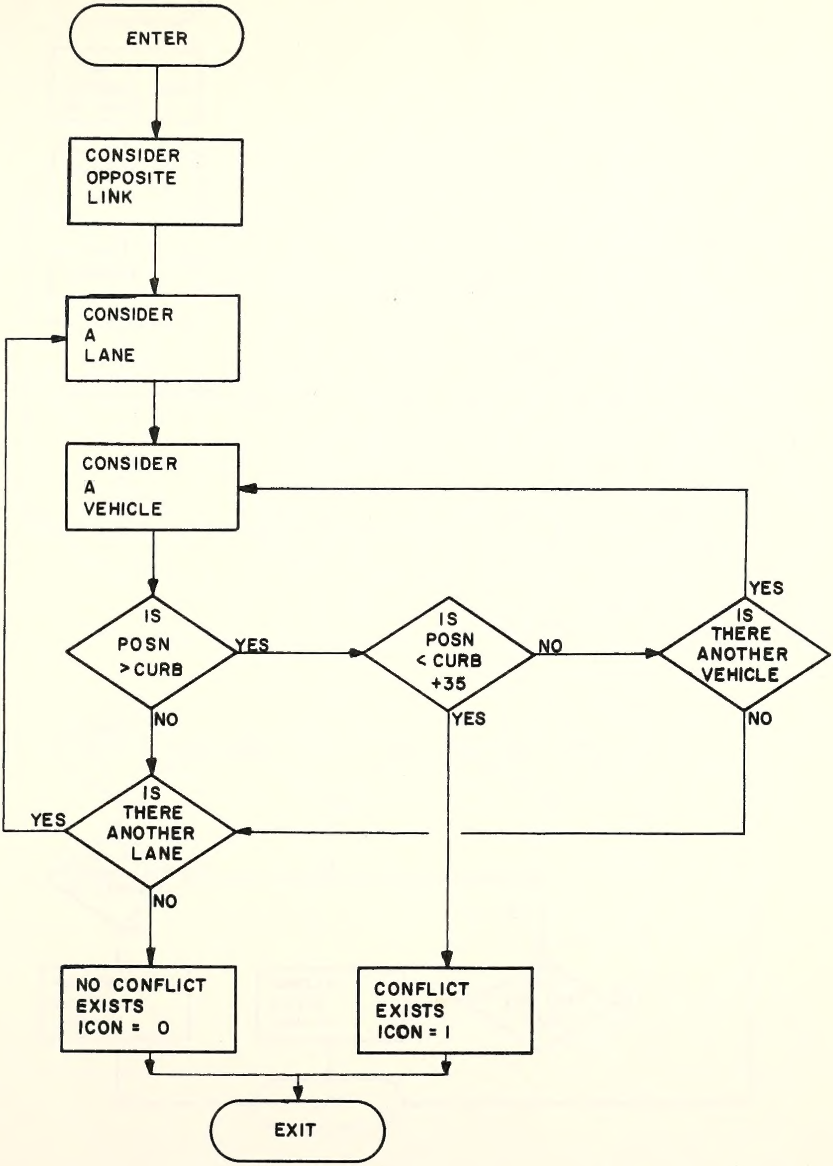


FIGURE 29
LEFT-TURN CONFLICT ROUTINE (CONLT)

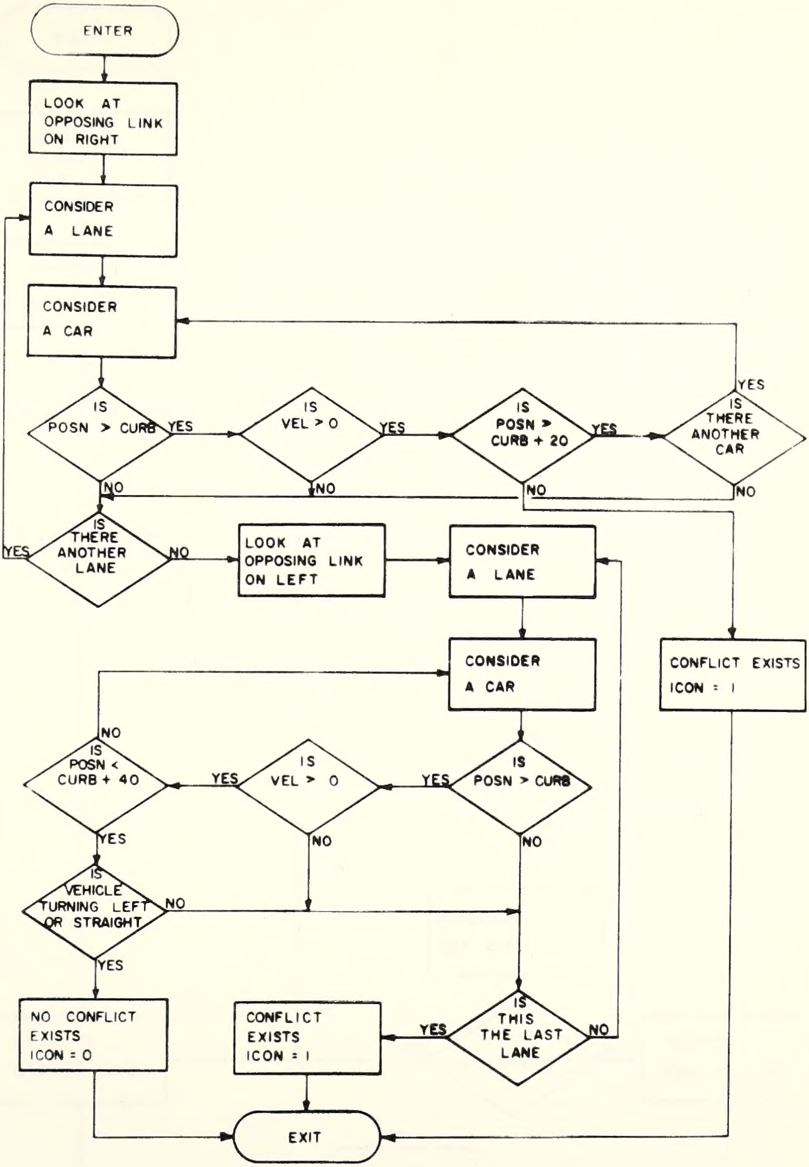


FIGURE 30

CROSS STREET CONFLICT ROUTINE (CONST)

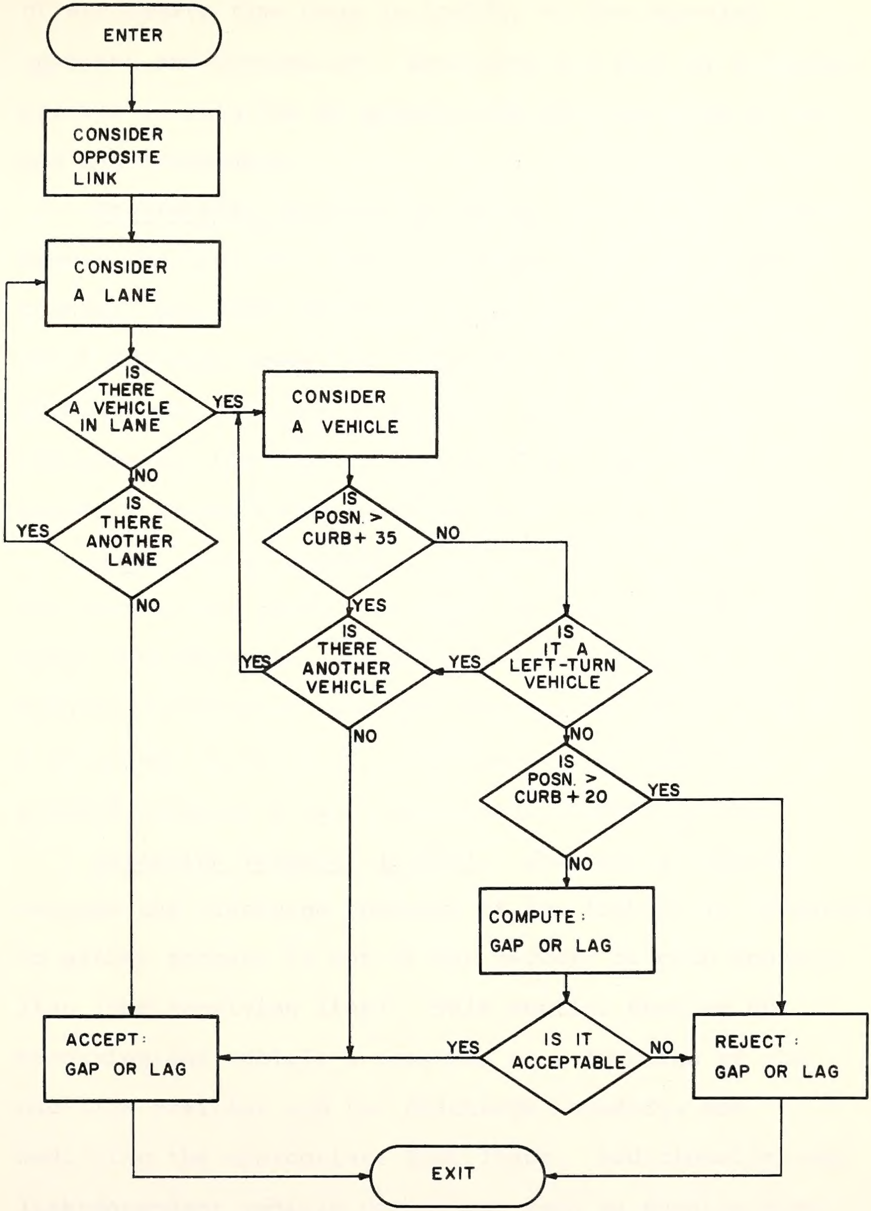


FIGURE 31

LAG & GAP ACCEPTANCE ROUTINE (GAP)

of successive time gaps in traffic on the opposing approach are determined. Decisions are made by a random process in relation to probability distributions of gap and lag acceptance.

Statistical Bookkeeping (BOOK). Each time a vehicle leaves its link this routine updates the applicable counters and sums, including total vehicles processed, total distance travelled, total travel time, total delay encountered, and sums of squares associated with each. Additionally the vehicle is marked for link transfer, which is accomplished in the routine OUTPT.

Queue Growth (QUEUE). This routine determines if a new vehicle has joined the queue on any of the links under consideration. Only those links specified on the Program Instruction Card (see Appendix B) are considered. A frequency analysis of queue lengths on these links is printed whenever a full statistical summary occurs.

Interlink Transfer (OUTPT). Whenever a vehicle crosses the discharge boundary of its link it is necessary to either process it out of the network or onto another link (the receiving link). This routine does so by resetting the vehicle's position as a function of its old-link position and the discharge boundary, and modifying the appropriate link lists. Additionally some link-dependent vehicle variables, such as turning movement and target velocity, are reset randomly.

Statistical Summary (SUMARY). This routine controls the computation, formatting, and writing on the output file of the summary statistics of network performance. It is executed at the end of the simulation run and at intermediate intervals as specified on the Program Instruction Card (see Appendix B). The routines STIX3 and STIX 6 are called by SUMARY to compute means and standard deviations and to compute, format, and write frequency tables of travel times and queue lengths.

Simulation Output

The simulation program output is a detailed statistical tabulation of traffic characteristics in the network. The first section of output is a listing of input data supplied to the simulation program. Included are program parameters, link parameters, link geometry descriptors, traffic signal settings, turning probabilities, and traffic volumes.

The second section of output presents traffic operation data. Traffic magnitude data related to generated volumes, link exit volumes, and volumes traced along the specified route are reported, followed by statistics computed from the individual vehicles' performance. Values are presented for the overall system and for individual links. The following performance measurements are reported:

Total Vehicle Miles: the total distance travelled on the link by all vehicles during the study period. Bookkeeping occurs only when a vehicle leaves its link; the distance travelled by vehicles still on the link at the time of the summary, therefore, is not included in this tally.

Total Delay (Given in seconds): A vehicle's undelayed travel time is based on its target velocity and distance travelled. The delay encountered by the vehicle, then, is the difference between its actual and undelayed travel time. Total delay for a link is the sum of all vehicles' delays on that link.

Average Delay (Given in seconds): the total delay divided by the number of vehicles leaving the link. In this study Average System Delay is taken to be the primary measurement of system performance.

Delay Standard Deviation: the standard deviation associated with the above average delay.

Average Delay (Given in seconds/vehicle-mile): an extended form of the average delay. Its purpose is to facilitate the comparison of average delays among links.

Total Travel Time (Given in seconds): the travel time for all vehicles.

Average Travel Time (Given in seconds/vehicle): the average time required to traverse the link under consideration.

Average Speed (Given in miles per hour): the average speed achieved by all vehicles over the entire link or network.

The final sections of the output report contain frequency tables of queue lengths on the specified links and travel-times on the route links.

CHAPTER IV

THE STREET NETWORK

A portion of the street network of West Lafayette, Indiana, was chosen as the test network. It consists primarily of the streets surrounding the Purdue University campus.

The major elements of the network are one major east-west arterial (S.R. 26), one major north-south arterial (U.S. 231), and three other north-south cross streets. The network thus defined includes sixteen signalized intersections. A street map of the network is shown in Figure 32.

A schematic diagram of the network is shown in Figure 33 with links numbered as required for use in the simulation model. The network consists of three types of links:

1. Input links that feed traffic into the network (links 60 through 84);

2. Network links that carry traffic from intersection to intersection within the network (links 26 through 59); and

3. Output links that remove traffic from the network (links 1 through 25).

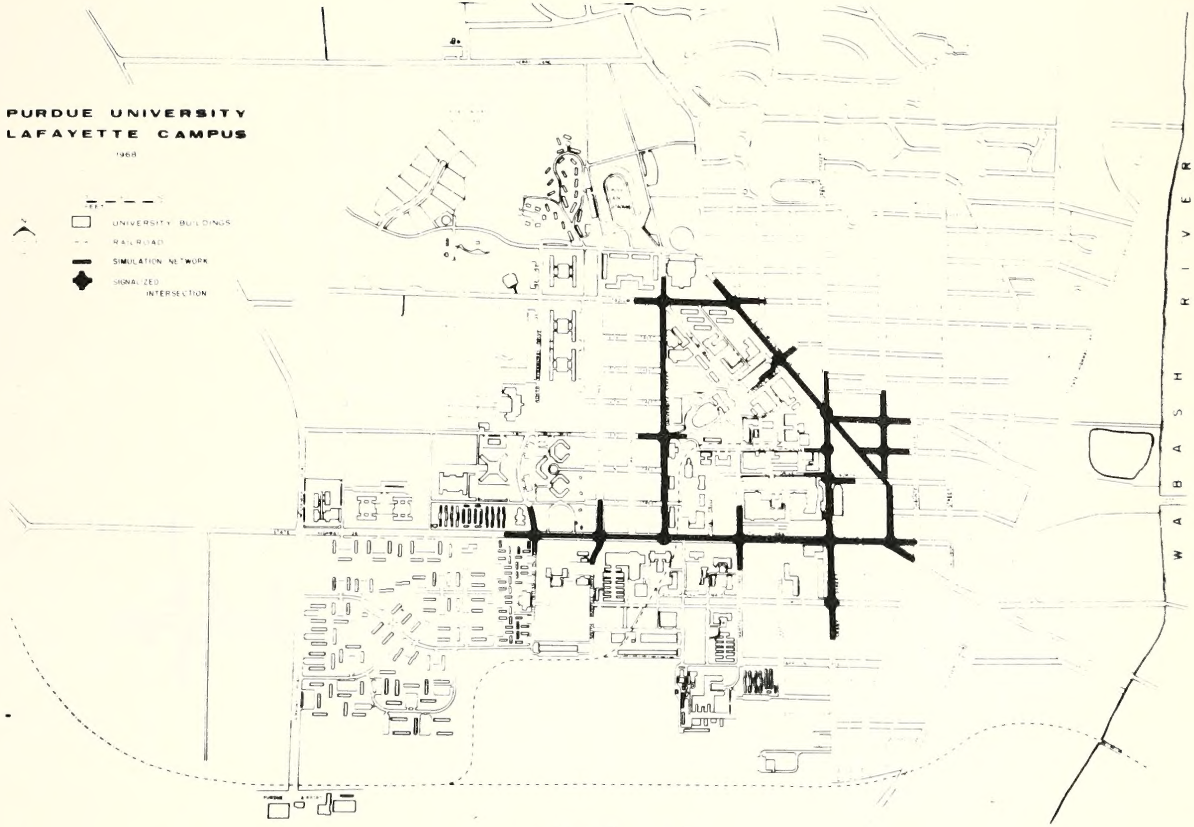


FIGURE 32

STREET MAP OF THE TEST NETWORK

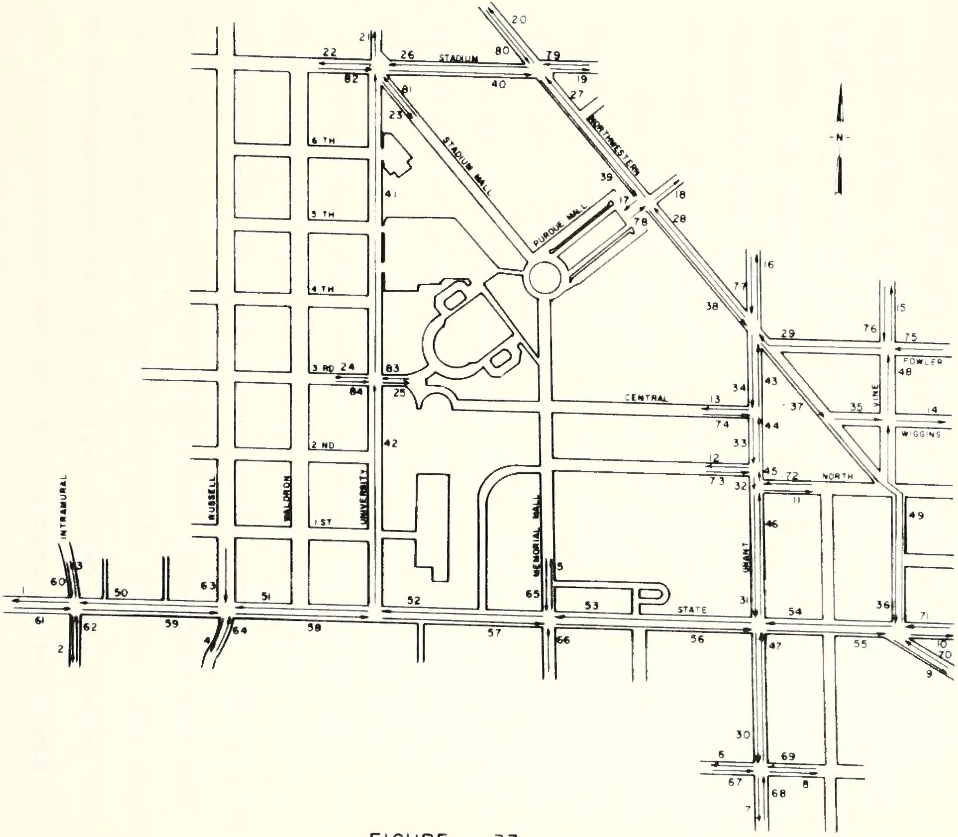


FIGURE 33

SCHEMATIC DIAGRAM OF THE TEST NETWORK

The SIGNET model does not accommodate minor streets not controlled by traffic signals. Two unusual characteristics of the network should be noted, however. First, the intersection of Northwestern Avenue and Wiggins Street is in actuality unsignalized. Because a major division of flow occurs at this point, however, a signal was assumed for use in the model which always indicated green. Thus no delay was caused to traffic but the required division was achieved.

Second, the intersection of Grant Street and North Street is such that the two east-west legs (North Street) are offset from each other. This situation is represented by employing two dummy links which form the offset. As in the intersection of Northwestern and Wiggins, the signals on these dummy links are always green.

Traffic signals in the network can be divided into three distinct systems. Those on State Street are interconnected and use a 65 second cycle length. The signal at Grant Street and Wood Street is also interconnected with this group and shares its cycle length. The signals on Northwestern Avenue, Wiggins Street, and Fowler Avenue are interconnected and use a 60 second cycle length. Finally, those city-operated signals at Stadium Avenue and University Street, Third Street and University Street, Grant Street and Central Avenue, and Grant Street and North Street operate on a variety of cycle lengths,

ranging from 55 to 70 seconds. There is no interconnection for these signals. All of the above cycle lengths are in effect during the study period, 4:00 p.m. to 5:30 p.m. The traffic signal settings are summarized in Appendix C.

For purposes of both simulation and signalization this network is considered to be independent of any other signals.

Information on the traffic characteristics of the network was required for two distinct phases of this study: (1) the validation of the simulation model; and (2) the application of the simulation program in testing alternative signal timing schemes. Common to both phases was the need for turning movement and lane distribution data. Traffic volumes were measured separately for each phase. The collection of travel time data is dealt with in Chapter 5, Model Testing and Validation.

Turning probabilities and lane distributions associated with each input and network link and input volumes associated with each input link are shown in Appendix C for average conditions during the study period, 4:00 to 5:30 p.m.

CHAPTER V

MODEL TESTING AND VALIDATION

The final phase in the development of the SIGNET model was testing and validation. It was desired to determine if the model represents real-world conditions with sufficient accuracy and if it is properly sensitive to changes in various input parameters.

The first phase of the sensitivity analysis was to compare travel times produced by the simulation model with equivalent travel times obtained in the field. It was presumed that a close correspondence between these times would indicate that the overall model was accurate. Individual portions of the model could then be tested in subsequent phases of the sensitivity analysis.

Actual travel times were obtained for each link on a route through the network between 4:00 and 4:45 p.m. using the floating-car method of driving. Input volumes were determined on a 15-minute basis for the same period of time. Due to limited resources it was possible to travel the route only 3 times during each 15-minute period, thereby yielding only 3 sets of travel time data for each set of input volume data.

Four simulation runs were made using different seeds for the random number generator and the input volumes described above. Histograms were made for each link in the route relating simulated travel times of vehicles traveling the route to the average frequency of occurrence. The actual travel times were then superimposed on these histograms to facilitate comparison. Several histograms are shown in Figures 34, 35, and 36.

Ideally the distribution of actual travel times should match the distribution of simulated travel times; if these distributions did match closely, it would be difficult to reject the simulation results as invalid. If the number of observations used in estimating both distributions were large it would be possible to apply standard statistical techniques in analyzing this goodness of fit. In this case, however, only 3 observations of actual travel times were available for each comparison. Standard statistical techniques break down under these conditions, making a qualitative analysis necessary.

Examination of the histograms reveals several general characteristics of the distributions. In all cases, the actual travel times fall within the distributions of simulated travel times, showing that the simulated travel times approximate actual travel times. At this point two cases arise in the comparison, both being dependent upon the relationship of demand volume and capacity. For those links on which demand volume does not approach capacity

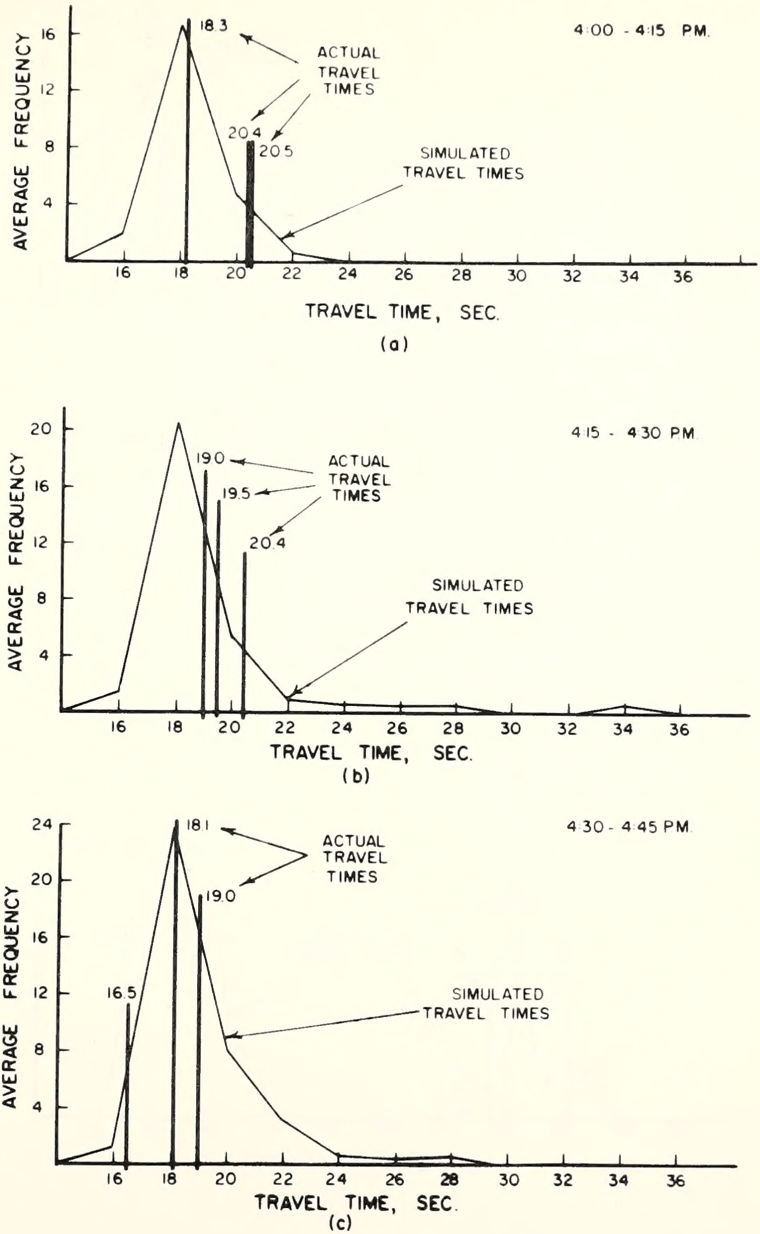
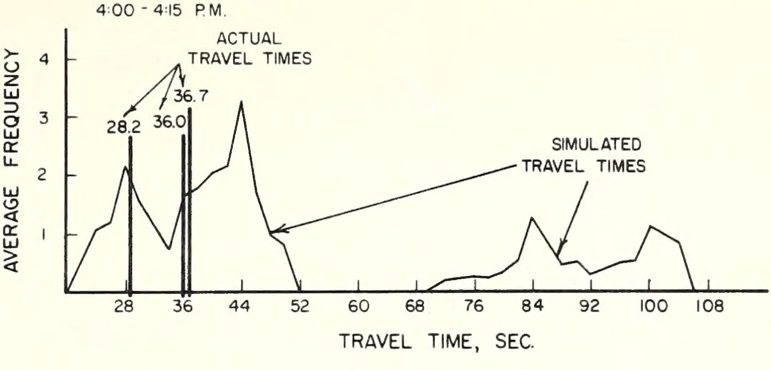
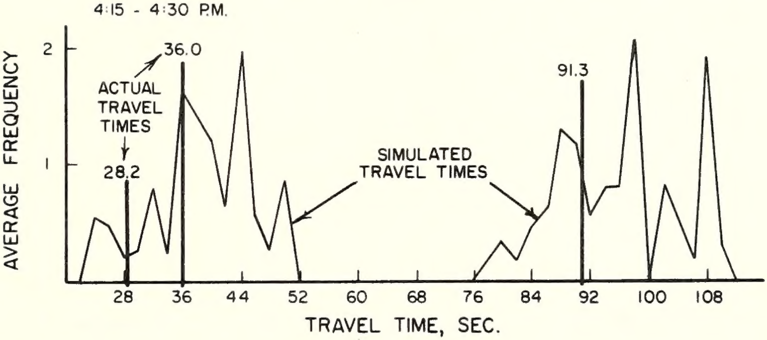


FIGURE 34

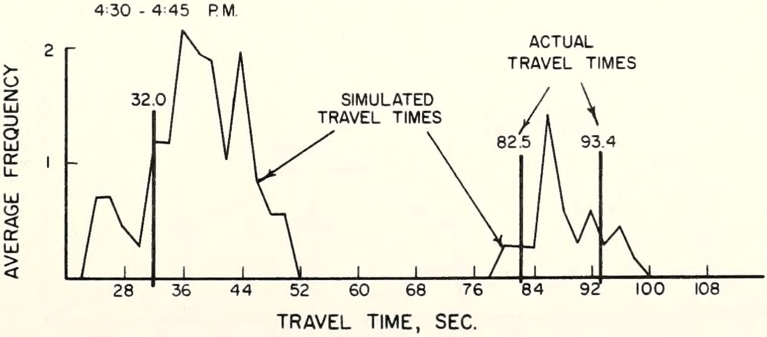
COMPARISON OF TRAVEL TIMES, LINK 28



(a)



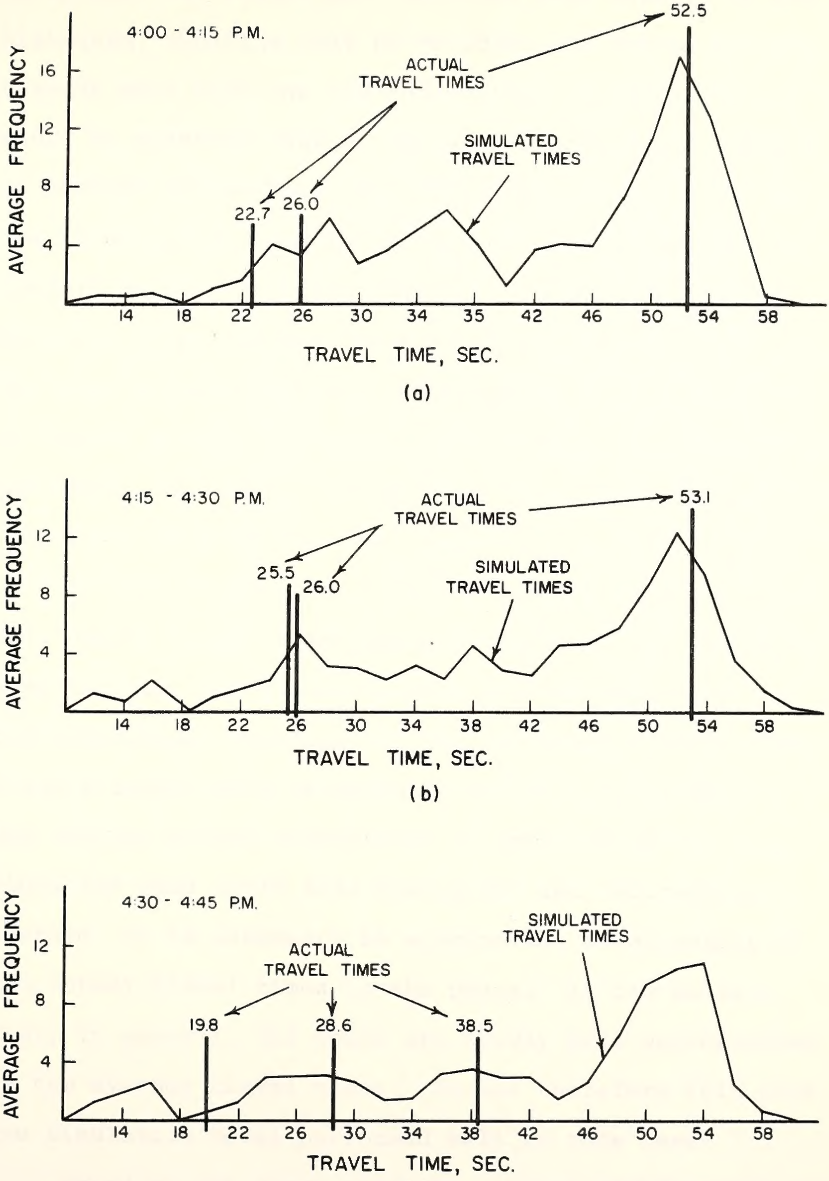
(b)



(c)

FIGURE 35

COMPARISON OF TRAVEL TIMES, LINK 56



(a)

(b)

(c)

FIGURE 36

COMPARISON OF TRAVEL TIMES, LINK 59

(i.e. Link 28, Figure 34), only one peak appears in the histogram, implying that no vehicles are forced to wait through more than one red indication. In this case it would be expected that an average vehicle would achieve an average travel time over the link; therefore, since the floating car used in the travel time collection is assumed to be an average vehicle, it would be expected that the actual travel times would fall approximately on the mean of the simulated travel times. It can be seen in Figure 34 that the actual travel times did approximate the mean. The overall model would seem to be performing relatively well in this instance.

On some links (i.e. Links 56 and 59, Figures 35 and 36), demand volume does approach capacity. In this case, some vehicles are forced to wait through more than one red indication, resulting in the two peaks shown. Under these circumstances it would be unrealistic to expect the average actual travel time to approximate the mean, since the mean could fall during the red indication. Instead, it is necessary to examine the relationship of the actual travel times to the peaks. It can be seen that, in general, the peaks are fairly well approximated by the average travel times. It was therefore felt that the simulation model performed well in this case.

Based on the above analysis, it was decided that the overall model was an acceptable one. That is, the entire

model working as a whole produced acceptable results. It remained to be shown, however, that individual portions of the model were also working. This phase of the testing is described below.

The objective of the second phase of the testing was to determine the sensitivity of the model output to changes in various input parameters. An inherent characteristic of computer simulation, the ability to control all input parameters precisely, enables the investigator to change only one parameter and thereby to determine its influence on the output. This phase of the sensitivity analysis had practical usefulness for two reasons:

1. It provided additional data which could be used in confirming the reasonableness of the model. The direction of change in the output caused by modifying a parameter could be examined to see if it was logical, and the magnitude of the change could be checked for reasonableness.

2. It indicated whether particular portions of the model logic were indeed operating. If a parameter modification produced no change where one was expected, there was good evidence of logic flaws in the model.

Sensitivity tests were made on 9 separate effects involving 7 input parameters. For all of the effects an attempt was made to test throughout the range of magnitudes

which might reasonably occur in real life. The simulation run of 4:00 to 4:15 p.m. was used throughout the sensitivity analysis as a base for comparison purposes. The following effects were tested:

1. The effect of variation in input volume on average delay. Greater volumes were expected to increase delay.

2. The effect of variation in input volume on average travel time and average speed. Greater volumes were expected to increase average travel time and thus decrease average speed.

3. The effect of variation in input volume on average queue length. Greater volumes were expected to increase average queue length.

4. The effect of variation in target velocity on average travel time and average speed. Greater target velocity was expected to decrease average travel time and thus increase average speed.

5. The effect of variation in effective length on avrage travel time and average speed. Greater effective length was expected to increase average travel time and thus decrease average speed.

6. The effect of variation in the truck percentage on average travel time and average speed. A higher truck percentage was expected to increase travel time and thus decrease average speed.

7. The effect of variation in the maximum desired acceleration on average travel time and average speed. A higher maximum desired acceleration was expected to decrease average travel time and thus increase average speed.

8. The effect of variation in the characteristic speed (a_o) on average travel time and average speed. A higher characteristic speed was expected to reduce average travel time and thus increase average speed.

9. The effect of variation in the proportionality coefficient (K) on average travel time and average speed. A higher proportionality coefficient was expected to result in decreased travel time and thus increased average speed.

Detailed results of the sensitivity tests are exhibited in Figures 37 through 52 and Tables 6 through 21. Four runs were made for each value of the input parameters. These provided a good data base upon which to build the Analysis of Variance for each effect.

It was desirable to know not only that the simulation output varied with changes in the input parameters, but also what form the variation took (i.e., if the response curve was linear, quadratic, or cubic in nature). If the output variation levelled out at some point this response could then be compared with expectations, thereby providing added evidence of the

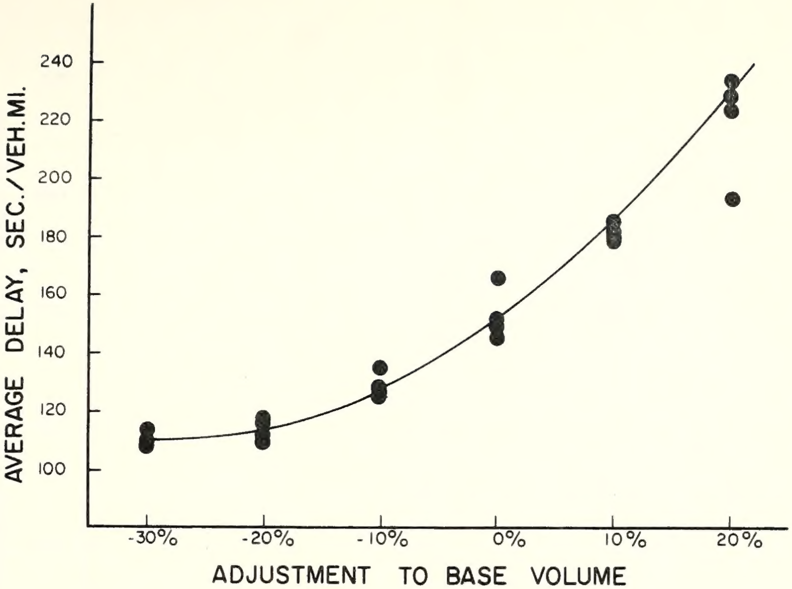


FIGURE 37

EFFECT OF VOLUME ON AVERAGE DELAY

TABLE 6
ANALYSIS OF VARIANCE
EFFECT OF VOLUME ON AVERAGE DELAY

SOURCE OF VARIATION	SUM OF SQUARES	D.F.	MEAN SQUARE	F RATIO
AMONG TREATMENTS	43631.71	3	14543.90	78.96 *
LINEAR EFFECT	39856.58	1	39856.58	216.3 *
QUADRATIC EFFECT	3747.26	1	3747.26	20.4 *
CUBIC EFFECT	27.86	1	27.86	0.161
EXPERIMENTAL ERROR	3683.79	20	184.19	
TOTAL	47315.50	23		

* EFFECT IS SIGNIFICANT AT $\alpha = 0.10$.
CRITICAL REGION: $F > 2.97$

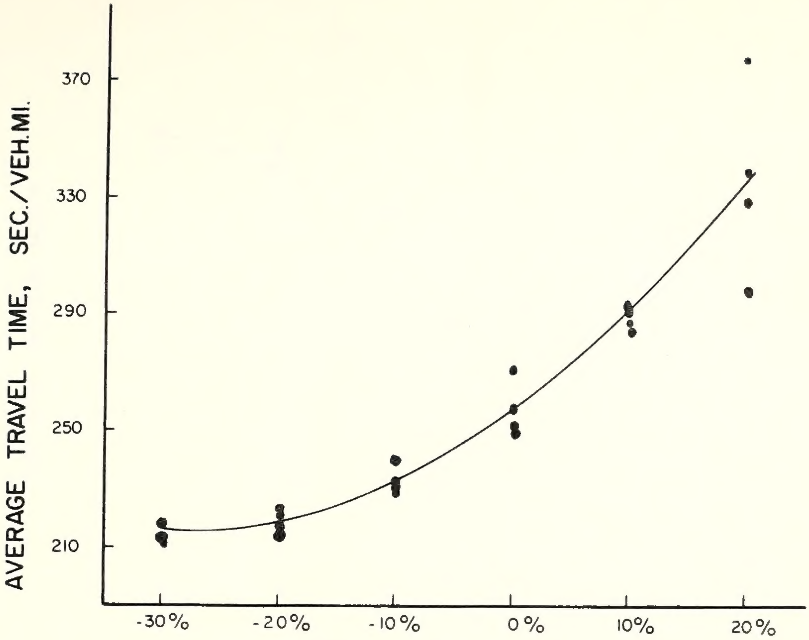


FIGURE 38

EFFECT OF VOLUME ON AVERAGE TRAVEL TIME

TABLE 7

ANALYSIS OF VARIANCE

EFFECT OF VOLUME ON AVERAGE TRAVEL TIME

SOURCE OF VARIATION	SUM OF SQUARES	D. F.	MEAN SQUARE	F RATIO
AMONG TREATMENTS	44707.17	3	14902.37	77.98 *
LINEAR EFFECT	40990.93	1	40990.93	213.9 *
QUADRATIC EFFECT	3715.69	1	3715.69	19.4 *
CUBIC EFFECT	0.501	1	0.501	0.003
EXPERIMENTAL ERROR	3822.04	20	191.10	
TOTAL	48529.16	23		

* EFFECT IS SIGNIFICANT AT $\alpha = 0.10$.

CRITICAL REGION: $F > 2.97$

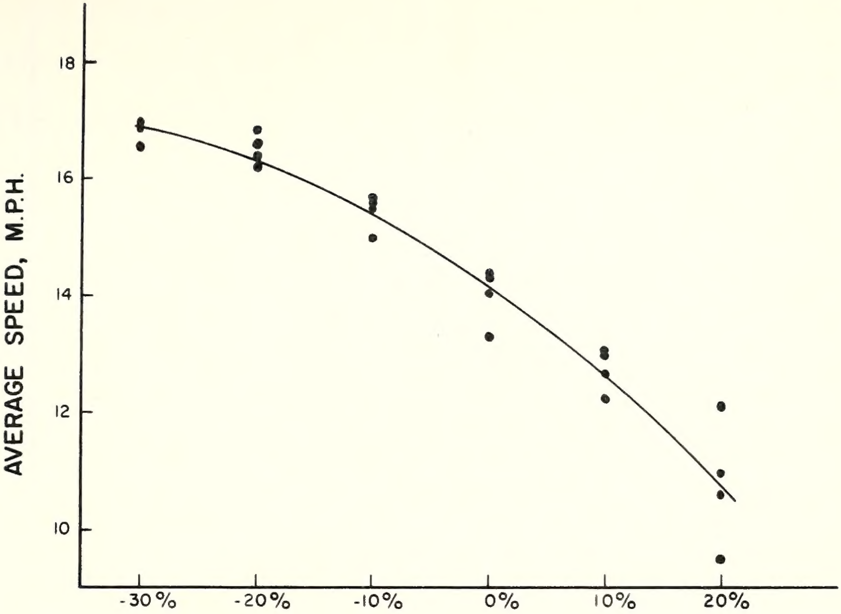


FIGURE 39

EFFECT OF VOLUME ON AVERAGE SPEED

TABLE 8

ANALYSIS OF VARIANCE

EFFECT OF VOLUME ON AVERAGE SPEED

SOURCE OF VARIATION	SUM OF SQUARES	D.F.	MEAN SQUARE	F RATIO
AMONG TREATMENTS	107.62	3	35.87	132.8 *
LINEAR EFFECT	103.66	1	103.66	383.3 *
QUADRATIC EFFECT	3.85	1	3.85	14.3 *
CUBIC EFFECT	0.10	1	0.10	0.38
EXPERIMENTAL ERROR	5.40	20	0.270	
TOTAL	113.02	23		

* EFFECT IS SIGNIFICANT AT $\alpha = 0.10$.

CRITICAL REGION: $F > 2.97$

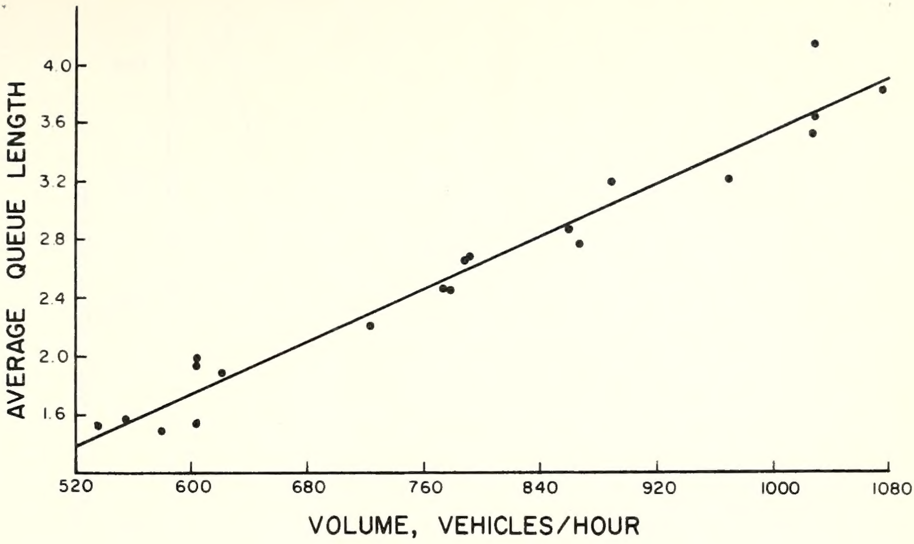


FIGURE 40

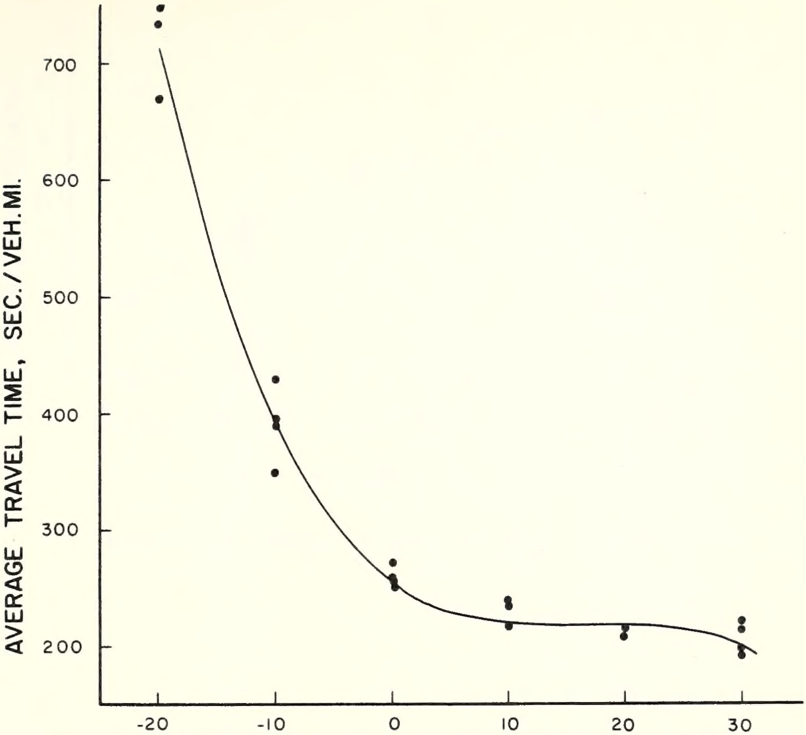
EFFECT OF VOLUME ON AVERAGE QUEUE LENGTH (LINK 57)

TABLE 9
ANALYSIS OF VARIANCE

EFFECT OF VOLUME ON AVERAGE QUEUE LENGTH (LINK 57)

SOURCE OF VARIATION	SUM OF SQUARES	D.F.	MEAN SQUARE	F RATIO
AMONG TREATMENTS	12.152	3	4.051	128.0 *
LINEAR EFFECT	12.142	1	12.142	384.5 *
QUADRATIC EFFECT	0.010	1	0.010	0.326
CUBIC EFFECT	0.0002	1	0.0002	0.005
EXPERIMENTAL ERROR	0.538	17	0.0317	
TOTAL	12.690	20		

* EFFECT IS SIGNIFICANT AT $\alpha = 0.10$.
CRITICAL REGION : $F > 3.03$



ADJUSTMENT TO BASE TARGET VELOCITY, M.P.H.

FIGURE 41

EFFECT OF TARGET VELOCITY ON AVERAGE TRAVEL TIME

TABLE 10
ANALYSIS OF VARIANCE

EFFECT OF TARGET VELOCITY ON AVERAGE TRAVEL TIME

SOURCE OF VARIATION	SUM OF SQUARES	D.F.	MEAN SQUARE	F RATIO
AMONG TREATMENTS	638599.4	3	212866.5	423.6 *
LINEAR EFFECT	435057.6	1	435057.6	865.8 *
QUADRATIC EFFECT	174491.2	1	174491.2	347.3 *
CUBIC EFFECT	29050.6	1	29050.6	57.8 *
EXPERIMENTAL ERROR	9547.2	19	502.5	
TOTAL	648146.6	22		

* EFFECT IS SIGNIFICANT AT $\alpha = 0.10$.
CRITICAL REGION: $F > 2.99$

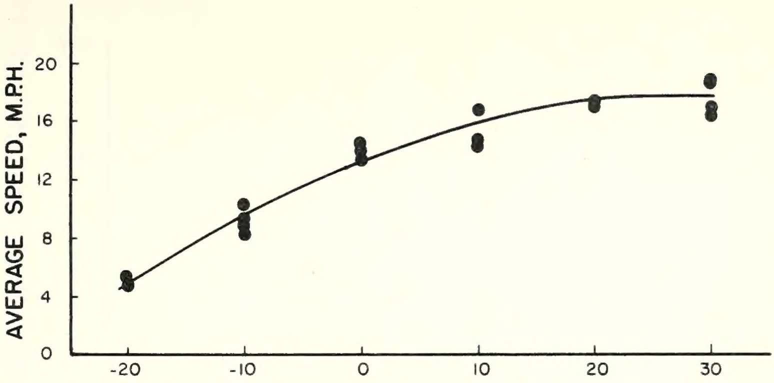


FIGURE 42

EFFECT OF TARGET VELOCITY ON AVERAGE SPEED

TABLE II

ANALYSIS OF VARIANCE

EFFECT OF TARGET VELOCITY ON AVERAGE SPEED

SOURCE OF VARIATION	SUM OF SQUARES	D. F.	MEAN SQUARE	F RATIO
AMONG TREATMENTS	422.56	3	140.85	233.6 *
LINEAR EFFECT	377.64	1	377.64	626.2 *
QUADRATIC EFFECT	44.91	1	44.91	74.5 *
CUBIC EFFECT	0.008	1	0.008	0.014
EXPERIMENTAL ERROR	11.46	19	0.603	
TOTAL	434.02	22		

* EFFECT IS SIGNIFICANT AT $\alpha = 0.10$.

CRITICAL REGION: $F > 2.99$

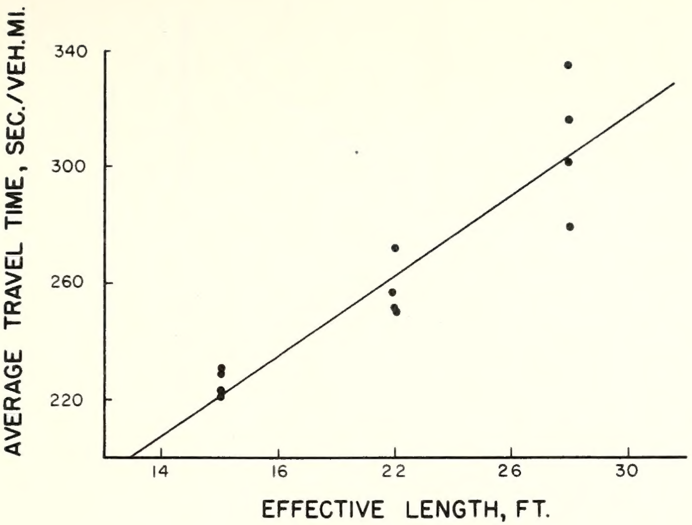


FIGURE 43

EFFECT OF EFFECTIVE LENGTH ON AVERAGE TRAVEL TIME

TABLE 12

ANALYSIS OF VARIANCE

EFFECT OF EFFECTIVE LENGTH ON AVERAGE TRAVEL TIME

SOURCE OF VARIATION	SUM OF SQUARES	D.F.	MEAN SQUARE	F RATIO
AMONG TREATMENTS	13631.18	2	6815.59	30.5 *
LINEAR EFFECT	13416.86	1	13416.86	60.2 *
QUADRATIC EFFECT	214.32	1	214.32	0.96
EXPERIMENTAL ERROR	2008.72	9	223.19	
TOTAL	15639.90	11		

* EFFECT IS SIGNIFICANT AT $\alpha = 0.10$.
CRITICAL REGION: $F > 3.23$

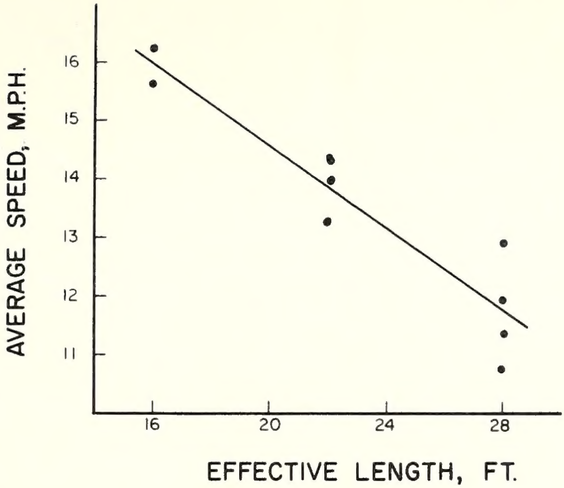


FIGURE 44

EFFECT OF EFFECTIVE LENGTH ON AVERAGE SPEED

TABLE 13

ANALYSIS OF VARIANCE

EFFECT OF EFFECTIVE LENGTH ON AVERAGE SPEED

SOURCE OF VARIATION	SUM OF SQUARES	D.F.	MEAN SQUARE	F RATIO
AMONG TREATMENTS	35.363	2	17.681	43.9 *
LINEAR EFFECT	35.322	1	35.322	87.7 *
QUADRATIC EFFECT	0.041	1	0.041	0.101
EXPERIMENTAL ERROR	3.625	9	0.403	
TOTAL	38.988	11		

* EFFECT IS SIGNIFICANT AT $\alpha = 0.10$.
CRITICAL REGION: $F > 3.36$

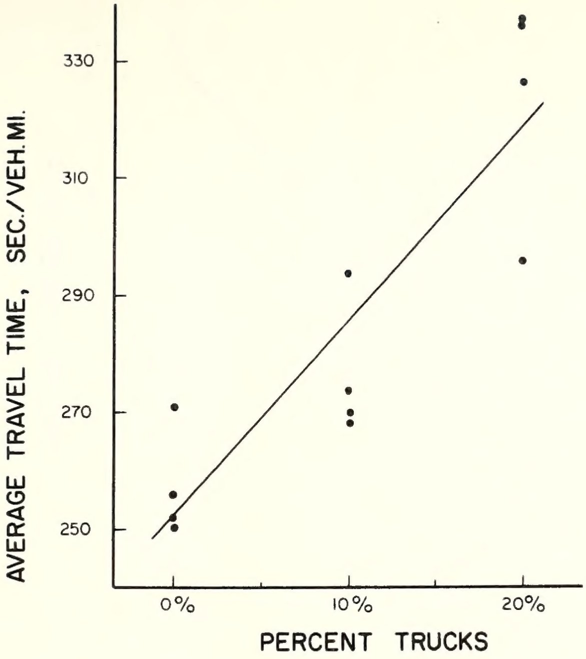


FIGURE 45

EFFECT OF TRUCKS ON AVERAGE TRAVEL TIME

TABLE 14

ANALYSIS OF VARIANCE

EFFECT OF TRUCKS ON AVERAGE TRAVEL TIME

SOURCE OF VARIATION	SUM OF SQUARES	D.F.	MEAN SQUARE	F RATIO
AMONG TREATMENTS	9406.99	2	4703.49	23.4 *
LINEAR EFFECT	8851.82	1	8851.82	44.1 *
QUADRATIC EFFECT	555.17	1	555.17	2.76
EXPERIMENTAL ERROR	1810.03	9	201.11	
TOTAL	11217.02	11		

* EFFECT IS SIGNIFICANT AT $\alpha = 0.10$.
CRITICAL REGION: $F > 3.36$

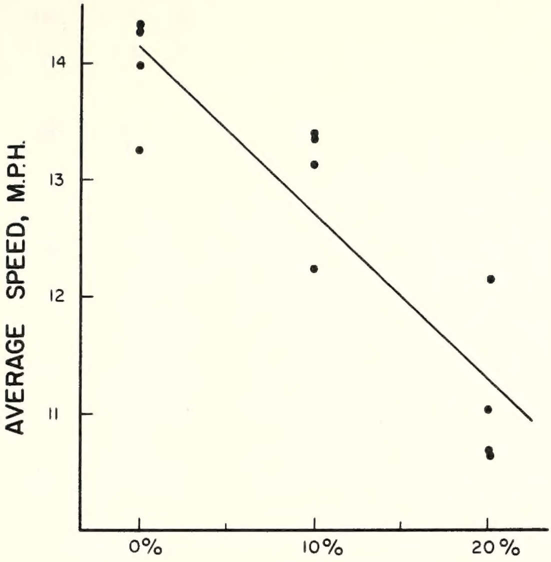


FIGURE 46

EFFECT OF TRUCKS ON AVERAGE SPEED

TABLE 15
ANALYSIS OF VARIANCE
EFFECT OF TRUCKS ON AVERAGE SPEED

SOURCE OF VARIATION	SUM OF SQUARES	D.F.	MEAN SQUARE	F RATIO
AMONG TREATMENTS	16.815	2	8.408	24.5 *
LINEAR EFFECT	16.188	1	16.188	47.2 *
QUADRATIC EFFECT	0.627	1	0.627	1.83
EXPERIMENTAL ERROR	3.084	9	0.343	
TOTAL	19.899	11		

* EFFECT IS SIGNIFICANT AT $\alpha = 0.10$.
CRITICAL REGION: $F > 3.36$

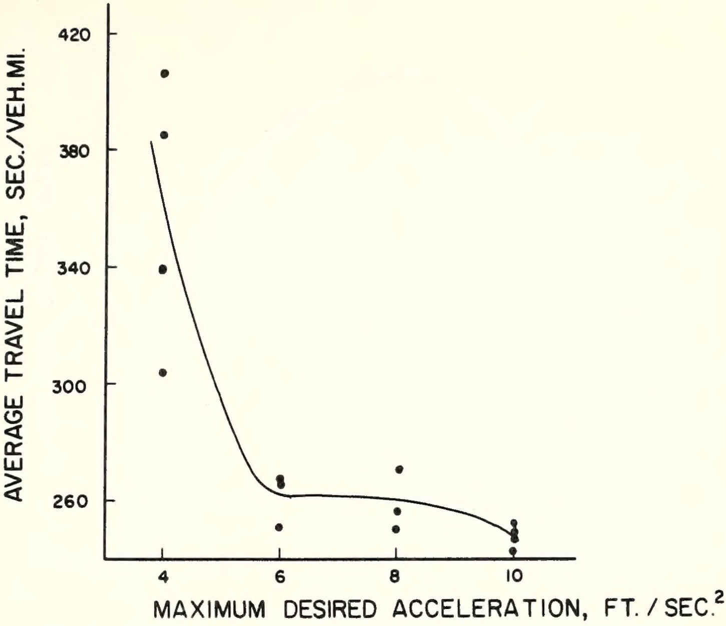


FIGURE 47

EFFECT OF MAXIMUM DESIRED ACCELERATION ON AVERAGE TRAVEL TIME

TABLE 16
ANALYSIS OF VARIANCE
EFFECT OF MAXIMUM DESIRED ACCELERATION ON AVERAGE TRAVEL TIME

SOURCE OF VARIATION	SUM OF SQUARES	DEGREES OF FREEDOM	MEAN SQUARE	F RATIO
AMONG TREATMENTS	32086.9	3	10695.6	19.1 *
LINEAR EFFECT	22890.4	1	22890.4	40.9 *
QUADRATIC EFFECT	7399.0	1	1399.0	13.2 *
CUBIC EFFECT	1797.5	1	1797.5	3.21 *
EXPERIMENTAL ERROR	6714.8	12	559.6	
TOTAL	38801.7	15		

* EFFECT IS SIGNIFICANT AT $\alpha = 0.10$.

CRITICAL REGION: $F > 3.18$

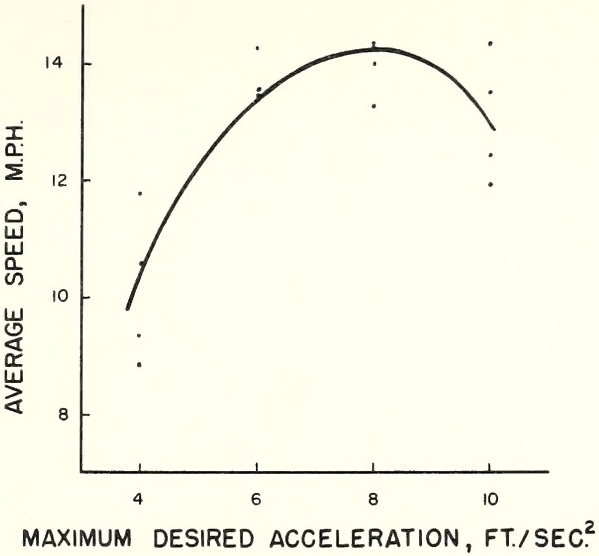


FIGURE 48

EFFECT OF MAXIMUM DESIRED ACCELERATION ON AVERAGE SPEED

TABLE 17

ANALYSIS OF VARIANCE,

EFFECT OF MAXIMUM DESIRED ACCELERATION ON AVERAGE SPEED

SOURCE OF VARIATION	SUM OF SQUARES	D.F.	MEAN SQUARE	F RATIO
AMONG TREATMENTS	36.93	3	12.31	14.3 *
LINEAR EFFECT	16.52	1	16.52	19.2 *
QUADRATIC EFFECT	19.56	1	19.56	22.8 *
CUBIC EFFECT	.85	1	.85	.99
EXPERIMENTAL ERROR	10.31	12	.859	
TOTAL	47.24	15		

* EFFECT IS SIGNIFICANT AT $\alpha = 0.10$

CRITICAL REGION: $F > 3.18$

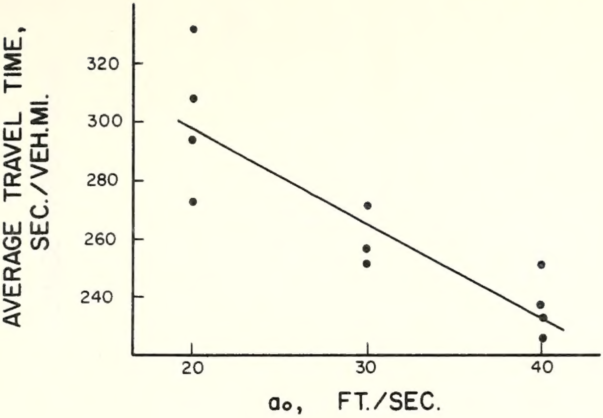


FIGURE 49
EFFECT OF CHARACTERISTIC SPEED (a_o) ON AVERAGE TRAVEL TIME

TABLE 18

ANALYSIS OF VARIANCE

EFFECT OF CHARACTERISTIC SPEED (a_o) ON AVERAGE TRAVEL TIME

SOURCE OF VARIATION	SUM OF SQUARES	D.F.	MEAN SQUARE	F RATIO
AMONG TREATMENTS	8787.66	2	4393.03	15.9 *
LINEAR EFFECT	8432.46	1	8432.46	30.4 *
QUADRATIC EFFECT	355.20	1	355.20	1.28
EXPERIMENTAL ERROR	2488.82	9	276.54	
TOTAL	11276.48	11		

* EFFECT IS SIGNIFICANT AT $\alpha = 0.10$.

CRITICAL REGION: $F > 3.23$

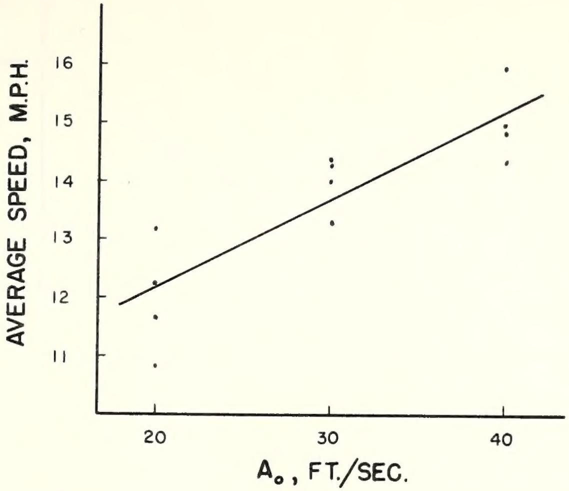


FIGURE 50

EFFECT OF CHARACTERISTIC SPEED (A_o) ON AVERAGE SPEED

TABLE 19

ANALYSIS OF VARIANCE, EFFECT OF CHARACTERISTIC SPEED ON AVERAGE SPEED

SOURCE OF VARIATION	SUM OF SQUARES	D. F.	MEAN SQUARE	F RATIO
AMONG TREATMENTS	18.909	2	9.455	16.8 *
LINEAR EFFECT	18.301	1	18.301	32.5*
QUADRATIC EFFECT	.608	1	.608	1.08
EXPERIMENTAL ERROR	5.073	9		
TOTAL	23.982	11		

* EFFECT IS SIGNIFICANT AT $\alpha = 0.10$
CRITICAL REGION: $F > 3.36$

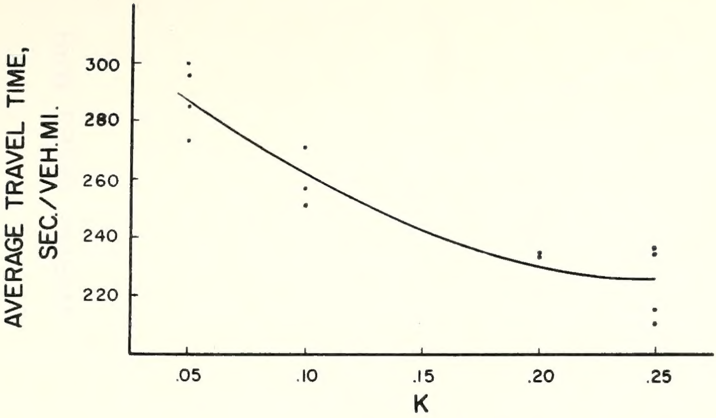


FIGURE 51

EFFECT OF PROPORTIONALITY COEFFICIENT (K) ON AVERAGE TRAVEL TIME

TABLE 20

ANALYSIS OF VARIANCE, EFFECT OF PROPORTIONALITY COEFFICIENT ON AVERAGE TRAVEL TIME

SOURCE OF VARIATION	SUM OF SQUARES	D.F.	MEAN SQUARE	F RATIO
AMONG TREATMENTS	9601.90	3	3200.63	31.20*
LINEAR EFFECT	9031.83	1	9031.83	88.00*
QUADRATIC EFFECT	437.65	1	437.65	4.27*
CUBIC EFFECT	132.42	1	132.42	1.29
EXPERIMENTAL ERROR	1231.04	12	102.59	
TOTAL	10832.93	15		

* EFFECT IS SIGNIFICANT AT $\alpha = 0.10$

CRITICAL REGION: $F > 3.18$

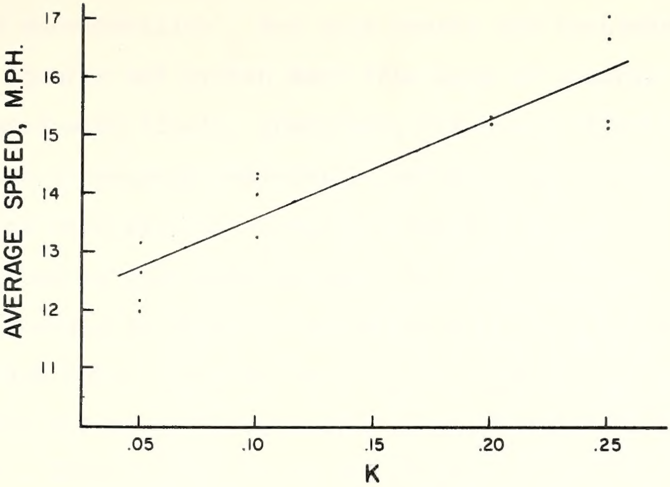


FIGURE 52

EFFECT OF PROPORTIONALITY COEFFICIENT (K) ON AVERAGE SPEED

TABLE 21
ANALYSIS OF VARIANCE,
EFFECT OF PROPORTIONALITY COEFFICIENT ON AVERAGE SPEED

SOURCE OF VARIATION	SUM OF SQUARES	D.F.	MEAN SQUARE	F RATIO
AMONG TREATMENTS	29.268	3	9.756	26.6 *
LINEAR EFFECT	28.375	1	28.375	77.3 *
QUADRATIC EFFECT	.518	1	.518	1.41
CUBIC EFFECT	.374	1	.374	1.02
EXPERIMENTAL ERROR	4.403	12		
TOTAL	33.671	15		

* EFFECT IS SIGNIFICANT AT $\alpha = 0.10$
CRITICAL REGION: $F > 3.18$

model's acceptability. For this reason the treatment sum of squares was broken down into sums of squares associated with linear, quadratic, and cubic effects using the orthogonal polynomials method. The results are shown for each test in Tables 6 through 21.

It can be seen from Figures 37, 38 and 39 that volume changes resulted in significantly less variation in average delay, average travel time, and average speed under low volume conditions than they did under high volumes. This flattening of the response curve was not unexpected, since under these low volume conditions delays caused by signals become much more significant than those caused by other vehicles.

Target velocity, the characteristics speed (a_o), the proportionality coefficient (K), and effective length are the primary parameters affecting queue discharge and thereby the total network performance. Decreases in target velocity cause slower discharge and lower speeds. (See Figures 41 and 42). As expected, increases are somewhat negated by the maximum desired acceleration (See Figures 47 and 48), since it puts a figurative ceiling on acceleration rates, and therefore causes a flattening of the response curve. Similarly, increases in the proportionality coefficient and the characteristic speed (Figures 49 through 52) cause faster queue discharge, while increased effective length (Figures 43 and 44) slows queue discharge

as was expected. Decreases in the maximum desired acceleration also slow queue discharge as expected. Studies by Greenshields (17) indicate that nearly all vehicles' starting accelerations are less than 6 ft/sec^2. Higher values for the maximum desired acceleration would therefore have little effect in slowing queue discharge. This expectation is parallelled in the response curve.

The way in which each of the above sensitivity tests performed as expected provided strong evidence that the individual portions of the SIGNET model were valid. These results, combined with the results of the travel time tests, served to substantially increase the author's confidence in the model.

CHAPTER VI

SIGNAL TIMING STUDIES

As stated in Chapter I, the ultimate objective of this research was to develop a tool which could be used to compare alternative traffic signal control schemes. The foregoing chapters have related to the development of this tool. This chapter is concerned with the application of the simulation program to testing these various schemes.

Several methods for optimizing signal systems were studied and reported on by Wagner, Barnes, and Gerlough (36). Among them were SIGOP, the British Combination Method, the Volume Priority Method, the Preferential Street Method, the Webster Method, and the Delay/Difference-of-Offset Method. Their findings indicate that application of any of these methods will produce significant upgrading of network performance.

Three of these techniques were used to develop alternative control schemes for this research. The Webster Method was used to determine optimal cycle lengths and splits for each intersection; the Delay/Difference-of-Offset Method was used to determine optimal offsets for each link; and the Preferential Street Method was used to determine the final offset plan for the network. Because full

descriptions of each of these methods appear elsewhere only a summary of each is presented below.

First published in a monograph in 1958 (38), the Webster method of determining optimum cycle lengths and splits for fixed time signals is rapidly becoming a standard method. Webster used a combination of theory and computer simulation to deduce a formula for average delay per vehicle at a signalized intersection. From that formula expressions were derived for the cycle lengths and splits which minimize total delay at the intersection. For a two-phase intersection:

$$C_0 = \frac{1.5L + 5}{1 - Y_A - Y_B}$$

where

C_0 = optimum cycle length, in seconds;

L = lost time per cycle, in seconds; and

Y_A and Y_B = the maximum ratios of single lane flow to saturation flow for phases A and B.

The splits should be established such that

$$\frac{GE_A}{GE_B} = \frac{Y_A}{Y_B}$$

where

GE_A and GE_B = effective green times of phases A and B, in seconds.

In the normal approach to strategic timing of signal networks the maximum cycle length determined above would be

taken as the network cycle length. This approach was used in the determination of one alternative, Alternative A, of this study. The difficulty with this practice is, however, that excessive cycle lengths may be used for large, virtually unrelated areas of the network. Wagner, Barnes, and Gerlough (36) found that under some conditions it is possible to mix cycle lengths and thereby optimize operation in two or more areas. Alternative B of this study was therefore designed to employ different cycle lengths on each of the two primary arterials, State Street and Northwestern Avenue. Simulation testing would be used to determine if the lack of fixed relationships between some adjacent signals adversely affected system performance.

First developed at the Road Research Laboratory in England for strategic optimization of offsets, the Delay/Difference-of-Offset Method was refined by Wagner, Gerlough, and Barnes (15). The traffic flows, the common cycle length, and the apportionment of green at each signal are given, and it is assumed that the delay to traffic along any link depends solely on the difference between the offsets of the signals at each end of the link. This offset optimization technique has perhaps the soundest logical base of the methods developed to date, because delay caused by queueing is taken under consideration and systematically reduced. A complete description of the technique may be found in N.C.H.R.P. Report 124 (36).

The Preferential Street Method is a relatively straightforward approach to combining the optimal offsets determined above into the final offset plan. It is one of the most widely used means for developing network timing plans, since the engineer uses his judgement and the information at hand to decide which streets are to be favored. The preferential streets must be selected in a manner which avoids closure of any portion of the network; in a grid network, this would entail selecting any number of parallel streets and one cross-street. The intersections along each preferential street are then assigned offsets as determined from some systematic offset determination technique, such as the Delay/Difference-of-Offset technique described above.

Before evaluating the network performance under each of the various signal control alternatives it was necessary to settle upon some criterion upon which to base this evaluation. Many criteria are available to the researcher, such as total travel time, mean travel time, maximum travel time, total system delay, mean system delay, maximum delay, total stopped delay, mean stopped delay, maximum stopped delay, mean queue length, maximum queue length, mean travel speed, and minimum travel speed.

Total or mean system delay has been widely accepted in theory as an appropriate measure of effectiveness. System delay is considered to be equal to the difference

between actual travel time and some theoretical optimum or undelayed travel time. Berry (5) assumed an undelayed trip to be one made at the average free flow speed. This assumption could be in error, however, since free flow speed does not necessarily correspond exactly with a vehicle's target velocity.

In the simulation model mean system delay is used as the primary measure of effectiveness of network performance. Undelayed travel time is calculated uniquely for each vehicle, based on the vehicle's target velocity, and delay is calculated as the difference between this undelayed travel time and the vehicle's actual travel time. Mean system delay is the average of all vehicles' delay.

A total of three different signal timing plans were tested for their effects on network performance. The existing condition was used as a basis for comparison, while Alternatives A and B were tested for any improvement in operation which they might produce. The simulation program was run three times for each condition using a different random number seed each time. The results are exhibited in Table 22.

Two statistical tests were performed which compared the operational effectiveness under Alternatives A and B with the effectiveness under the existing timing plan:

1. The variances of Alternatives A and B were tested for equality with the variance of the existing

TABLE 22

Average Delay Caused by Signal Timing Alternatives sec./veh.[1]

	Existing Condition	Alternative A	Alternative B
Run No. 1	19.47	22.65	19.21
Run No. 2	19.32	23.73	21.17
Run No. 3	20.07	21.09	21.26
Mean	19.62	22.49	20.55

---

[1] Delays are computed on a link basis. Therefore the delays shown are the average delays encountered in negotiating the length of one link.

condition; i.e., the hypotheses H_0: $\sigma_A^2 = \sigma_E^2$ and H_0: $\sigma_B^2 = \sigma_E^2$ were tested, as opposed to the alternatives H_1: $\sigma_A^2 = \sigma_E^2$ and H_1: $\sigma_B^2 = \sigma_E^2$ respectively. As defined by Ostle (30), the hypothesis H_0 can be accepted if

$$\chi^2 \ (\alpha/2, n-1) < \frac{\Sigma \ (X-\bar{X})^2}{\sigma_0^2} < \chi^2_{(1-\alpha/2, \ n-1)}.$$

Otherwise, H_0 must be rejected. In this case,

$$\frac{\Sigma \ (X - \bar{X})_A^2}{\sigma_E^2} = \frac{3.5145}{.1666} = 21.1$$

and

$$\frac{\Sigma \ (X - \bar{X})_B^2}{\sigma_E^2} = \frac{2.686}{.1666} = 16.1.$$

At the = 0.10 level, both values were greater than the critical χ^2, so the hypotheses were rejected. The variances, then, could not be considered equal.

2. The means of Alternatives A and B were tested for equality with the mean of the existing condition; i.e., the hypotheses H_0: $\mu_A = \mu_B$ and H_0: $\mu_B = \mu_E$ were tested, as opposed to the alternatives H_0: $\mu_A = \mu_E$ and H_0: $\mu_B = \mu_E$. A reasonably good approximate procedure for cases involving unequal variances is defined by Ostle. For the sample hypotheses H_0: $\mu_1 = \mu_2$, compute

$$t' = (\bar{X}_1 - \bar{X}_2) \; / \; (s_1^2/n_1 + s_2^2/n_2)^{1/2}$$

and reject if

$$t' \le -(w_1 t_1 + w_2 t_2) \; / \; (w_1 + w_2)$$

or if

$$t' \ge (w_1 t_1 + w_2 t_2) \; / \; (w_1 + w_2)$$

where

$$w_1 = s_1^2/n_1;$$

$$w_2 = s_2^2/n_2;$$

$$t_1 = t_{(1-\alpha/2,\ n_1-1)}; \text{ and}$$

$$t_2 = t_{(1-\alpha/2, n_2-1)}.$$

In this case,

$$t'_A = \frac{(22.488 - 19.619)}{\left(\frac{1.757}{3} + \frac{.1666}{3}\right)^{1/2}} = 3.58$$

and

$$t'_B = \frac{(20.546 - 19.619)}{\left(\frac{1.343}{3} + \frac{1.666}{3}\right)^{1/2}} = 1.31.$$

The weighted average of t_1 and t_2 was 2.920. Therefore H_0: $\mu_A = \mu_E$ was rejected while H_0: $\mu_B = \mu_E$ could not be rejected. In other words, the mean delay associated with Alternative A could not be considered equal to the mean delay associated with the existing condition. The mean delay of Alternative B, however, could not be said to be unequal to the mean delay of the existing condition.

It is not surprising that no significant improvement could be made on the existing condition, for that timing plan was derived over a period of time using both

analytical and empirical (i.e., field adjustment) techniques. It cannot be said that a design procedure incorporating simulation will consistently produce a better plan than a procedure using other techniques, for that is not the point of using simulation. Instead, simulation can help arrive at the optimal timing plan more quickly, primarily by providing an easily accessible medium for study and experimentation. This aspect of the project was demonstrated by the fact that it took only a few hours to develop, implement, and test the two alternatives in this study. The benefits of using simulation were further demonstrated by the fact that a significant difference was demonstrated between two plans, thereby indicating which should be implemented.

There is also a possibility that the techniques used to determine the timing plans were inadequate in some respects. The detection of such inadequacies was not in the scope of this project, nor would it be under normal design conditions. Inadequacies in the design techniques could contribute to the lack of improvement encountered in this instance, however; since simulation is merely a tool for choosing among the alternative timing plans, there is no way of overcoming such shortcomings using just simulation.

The foregoing was intended to give a somewhat detailed description of the use of the simulation program in the

design of signal timing plans. The steps required in this type of design may be summarized as follows:

1. Comprehensive data must be gathered on both the network geometry and traffic characteristics. Turning movements and input volumes are especially critical, and should be determined as precisely as resources allow.

2. The simulation program should be calibrated to the network by means of a travel time test and at least a cursory sensitivity analysis.

3. Alternative signal timing plans should be developed using systematic study and analysis techniques.

4. These alternative plans should be tested, using several simulation runs and standard statistical techniques to determine the significance of the results.

5. The final selection of one alternative should be made on the basis of one or more predefined criteria.

The above steps are predicated on the assumption that the goals, objectives, and constraints of the study have already been determined and that simulation is the most effective means available to achieve these objectives.

It is difficult at this point to determine how much time the use of simulation could save the traffic engineer, since it is unknown how long the field adjustment of a typical improvemen t would take. In this project preparation and coding of the input data took between 15 and 20 hours, although much of this data would also

normally be required for designing the timing plans. It is felt, therefore, that this figure compares favorably with the time required to undertake field adjustment. The possibility of using the simulation in studying other phases of traffic control, such as turning movement prohibition, parking regulation, and unbalanced lane operation, also tends to make the use of simulation more attractive. Such studies were not, however, attempted in this research.

CHAPTER VII

CONCLUSION

The overall accomplishments of this project conformed well with the original research plan. A simulation model, SIGNET, was developed to reproduce traffic flow under laboratory conditions. The program was fairly economical, achieving a 6.5 to 1 real-time to simulation-time ratio. A sensitivity analysis indicated that the model simulated traffic flows quite accurately.

The model was used to evaluate several alternative timing plans, and indicated by a significant difference in mean delay between two of these plans that it is indeed sensitive to variations in signal timing. No improvement could be made over the operation of the existing network. This does not necessarily indicate that the model itself is weak, however, because the traffic control theory used to derive the alternatives could be inadequate; under such circumstances no improvement could be expected. It was demonstrated that the incorporation of simulation into the design procedure could make that procedure quicker, at the same time resulting in the production of a more efficient timing plan. In effect this was the overall goal of the project.

The use of the SIGNET model is not restricted to signal timing studies. In fact, the effects of many other traffic engineering measures may be evaluated by varying the program inputs. Such measures could include turning movement prohibitions, parking restrictions, unbalanced lane operation, and one-way street operation. A comprehensive set of improvements to the network can thus be easily evaluated. This capability is important because traffic signal optimization cannot be considered to be the sole means of obtaining maximum network efficiency.

Of primary importance in future work involving SIGNET should be more intensive validation studies. The model was quite accurate in reproducing traffic conditions during the validation phase of this study; it would be desirable, however, to test the model much more if more comprehensive traffic data could be obtained. A constant monitoring of input volumes during collection of travel data would be especially beneficial in this regard, as would well defined turning probabilities.

The model itself could be somewhat improved also. Provision for pedestrian conflicts, right-turns on red, and traffic actuated signals would result in increased versatility. It is doubtful, however, that such additions would affect overall model performance substantially.

BIBLIOGRAPHY

BIBLIOGRAPHY

1. American Association of State Highway Officials, *A Policy on Geometric Design of Rural Highways*, Washington, D. C., 1965.

2. Bartle, R. M., Val Skoro, and Gerlough, D. L., "Starting Delay and Time Spacing of Vehicles Entering Signalized Intersection," *Effects of Traffic Control on Street Capacity*, Bulletin 112, Highway Research Board, pp. 33-41, 1956.

3. Bellis, W. R., "Capacity of Traffic Signals and Traffic Signal Timing," *Increasing Traffic Capacity of Arterial Streets*, Bulletin 271, Highway Research Board, pp. 45-67, 1960.

4. Benhard, F. G., "Simulation of a Traffic Intersection on a Digital Computer," Master of Science *Thesis*, University of California, Los Angeles, June, 1959.

5. Berry, D. S., "Field Measurement of Delay at Signalized Intersections," *Proceedings*, Highway Research Board, Vol. 35, pp. 505-522, 1956.

6. Blackman, A. R., and Crawford, A., "Driver Behavior During Amber Period of Traffic Lights: 6. Road Observations," Unpublished Research Note LN/106/ARB,AC, British Road Research Laboratory, May , 1962.

7. Brooks, W. D., *Designing Arterial Progressions Using a Digital Computer*, International Business Machines Corporation.

8. Carstens, R. L., "Some Traffic Parameters at Signalized Intersections," *Traffic Engineering*, Vol. 41, pp. 33-36, August, 1971.

9. Chang, A., *Synchronization of Traffic Signals in Grid Networks*, International Business Machines Corporation, San Jose Research Laboratory.

10. Dawson, R. F., and Chimini, L. A., "The Hyperlang Probability Distribution - A Generalized Traffic Headway Model," *Record 230*, Highway Research Board, pp. 1-14, 1968.

11. Gerlough, D. L., "Simulation of Freeway Traffic by an Electronic Computer," Proceedings, Highway Research Board, Vol. 35, pp. 543-547, 1956.

12. Gerlough, D. L., "Simulation of Traffic Flow," An Introduction to Traffic Flow Theory, Special Report 79, Highway Research Board, pp. 97-118, 1964.

13. Gerlough, D. L., and Wagner, F. A., "Improved Criteria for Traffic Signals at Individual Intersections, Interim Report," Report No. 3, National Cooperative Highway Research Program, 1964.

14. Gerlough, D. L., and Wagner, F. A., "Improved Criteria for Traffic Signals at Individual Intersections," Report No. 32, National Cooperative Highway Research Program, 1967.

15. Gerlough, D. L., Wagner, F. A., and Barnes, F. C., "Improved Criteria for Traffic Signal Systems on Urban Arterials," Report No. 73, National Cooperative Highway Research Program, 1969.

16. Goode, H. H., Polmar, C. H., and Wright, J. B., "The Use of a Digital Computer to Model a Signalized Intersection," Proceedings, Highway Research Board, Vol. 35, pp. 548-557, 1956.

17. Greenshields, B. D., "Some Time Space Relationships of Traffic in Urban Areas," Proceedings, Institute of Traffic Engineers, pp. 114-134, 1946.

18. Greenshields, B. D., Schapiro, D., and Erickson, E. L., Traffic Performance at Urban Street Intersections, Technical Report No. 1, Bureau of Highway Traffic, Yale University, 1947.

19. Hammond, H. F., "Report of Committee on Safe Approach Speeds at Intersections," Proceedings, Highway Research Board, Vol. 20, p. 657, 1940.

20. Heathington, K. W., and Rath, G. J., "Computer Simulation for Transportation Problems," Traffic Quarterly, April, 1968, pp. 271-281.

21. Herman, R., and Potts, R. B., "Single Lane Traffic Theory and Experiment," Theory of Traffic Flow, Elsevier, pp. 120-146, 1961.

11. Gerlough, D. L., "Simulation of Freeway Traffic by an Electronic Computer," *Proceedings*, Highway Research Board, Vol. 35, pp. 543-547, 1956.

12. Gerlough, D. L., "Simulation of Traffic Flow," *An Introduction to Traffic Flow Theory, Special Report 79*, Highway Research Board, pp. 97-118, 1964.

13. Gerlough, D. L. and Wagner, F. A., "Improved Criteria for Traffic Signals at Individual Intersections. Interim Report," *Report No. 3*, National Cooperative Highway Research Program, 1964.

14. Gerlough, D. L., and Wagner, F. A., "Improved Criteria for Traffic Signals at Individual Intersections," *Report No. 32, National Cooperative Highway Research Program*, 1967.

15. Gerlough, D. L., Wagner, F. A., and Barnes, M. C., "Improved Criteria for Traffic Signal Systems on Urban Arterials," *Report No. 73*, National Cooperative Highway Research Program, 1969.

16. Goode, H. H., Pollmar, C. H., and Wright, J. B., "The Use of a Digital Computer to Model a Signalized Intersection," *Proceedings*, Highway Research Board, Vol. 35, pp. 548-557, 1956.

17. Greenshields, B. D., "Some Time Space Relationships of Traffic in Urban Areas," *Proceedings*, Institute of Traffic Engineers, pp. 114-124, 1949.

18. Greenshields, B. D., Schapiro, D., and Ericksen, E. L., *Traffic Performance at Urban Street Intersections, Technical Report No. 1, Bureau of Highway Traffic*, Yale University, 1947.

19. Hammond, H. F., "Report of Committee on Safe Approach Speeds at Intersections," *Proceedings*, Highway Research Board, Vol. 20, p. 657, 1940.

20. Heathington, K. W., and Rath, G. J., "Computer Simulation for Transportation Problems," *Traffic Quarterly*, April, 1968, pp. 271-281.

21. Herman, R., and Potts, R. B., "Single Lane Traffic Theory and Experiment," *Theory of Traffic Flow*, Elsevier, pp. 120-146, 1961.

22. Herman, R., and Rothery, R. W., "Car Following and Steady State Flow," Proceedings of the International Symposium on Theory of Traffic Flow, London, pp. 1-11, 1963.

23. Hillier, J. A., "Glasgow's Experiment in Area Traffic Control," Traffic Engineering and Control, Vol. 7, No. 8, pp. 502-509, 1965. Appendix, Vol. 7, No. 9, pp. 569-571, 1966.

24. Institute of Traffic Engineers, Traffic Engineering Handbook, Washington, D. C., 1965.

25. Kell, J. H., "Analyzing Vehicular Delay at Intersections through Simulation," Theory of Traffic Flow, Bulletin 356, Highway Research Board, 1962.

26. Lewis, R. M., and Michael, H. L., "The Simulation of Traffic Flow to Obtain Volume Warrants for Intersection Control," Ph.D. Thesis, Purdue University, Lafayette, Indiana, 1962.

27. Lewis, R. M., "The Simulation of Vehicular Traffic at an Intersection on a Digital Electronic Computer," MCE Thesis, Rensselaer Polytechnic Institute, Troy, New York, May, 1956.

28. Little, J. D. C., Martin, B. V., and Morgan, J. T., "Synchronizing Signals for Maximal Bandwidth," Record 118, Highway Research Board, pp. 21-46, 1966.

29. Olson, P. L., and Rothery, R. W., "Driver Response to the Amber Phase of Traffic Signals," Traffic Engineering, Vol. 32, No. 5, pp. 17-21, 29, 1962.

30. Ostle, Bernard, Statistics in Research, Second Edition, The Iowa State University Press, Ames, Iowa, 1963.

31. "Progression," Unpublished Handout from CE 565, Traffic Engineering Operations and Controls, Purdue University, 1971.

32. Schwartz, Jesse G., An Urban Street Network Computer Simulation Model, Massachusetts Institute of Technology, Department of Civil Engineering Report R66-52, September, 1966.

33. Stark, M. C., "Computer Simulation of Traffic on Nine Blocks of a City Street," Theory of Traffic Flow, Bulletin 356, Highway Research Board, 1962.

34. Strickland, R. I., "Traffic Operation at Vehicle Tunnels," Proceedings, Highway Research Board, Vol. 33, p. 398, 1956.

35. Transportation Engineering Center, "Study of Electronic Devices as Traffic Aids," Report No. 202-1, Ohio State University, 1962.

36. Wagner, F. A., Barnes, F. C., and Gerlough, D. L., "Improved Criteria for Traffic Signal Systems in Urban Networks," Report No. 124, National Cooperative Highway Research Program, 1971.

37. Wagner, F. A., Barnes, F. C., Stirling, D. P., and Gerlough, D. L.,"Urban Arterial and Network Simulation," Planning Research Corporation Report R-926, prepared for U. S. Dept. of Commerce, Bureau of Public Roads, Office of Research and Development (PB 174629).

38. Webster, F. V., "Traffic Signal Settings," Road Research Technical Paper No. 39, London, Her Majesty's Stationery Office, 1958.

39. Wilson, E. E., "Deceleration Distances for High Speed Vehicles," Proceedings, Highway Research Board, Vol. 20, pp. 393-398, 1940.

40. Wong, S. Y., "Traffic Simulation with a Digital Computer," Proceedings of the Western Joint Computer Conference, San Francisco, California, pp. 92-94, February 7-9, 1956.

41. Yardeni, L. A., "Algorithms for Traffic Signal Control," IBM Systems Journal, Vol. 4, No. 2, 1965.

APPENDICES

Appendix A

PRESIG Program Details

The following is a complete description of the PRESIG (preliminary geometry determination) program inputs. All data are directly related to the geometry of the network, and are easily measured in the field. The function of PRESIG is to convert these data to data needed by the simulation program SIGNET, using the procedure described in Chapter III. A summary of the required variables and formats is given in Table A-2.

The physical characteristics required for each link are as follows:

$ITYPE_i$ = intersection type, numerical class.

$JTYPE_i$ = intersection type, alphabetical subclass.

The intersection types recognized by PRESIG are shown in Figure A-1 with their class (number) and subclass (letter) identification. Care must be taken that the intersection is considered from the same orientation as it would be by an approaching driver on the link. For the purposes of PRESIG, one-way streets toward an intersection are not considered by the other links of that intersection. Therefore when the intersection type is chosen, one-way legs toward the intersection are not included.

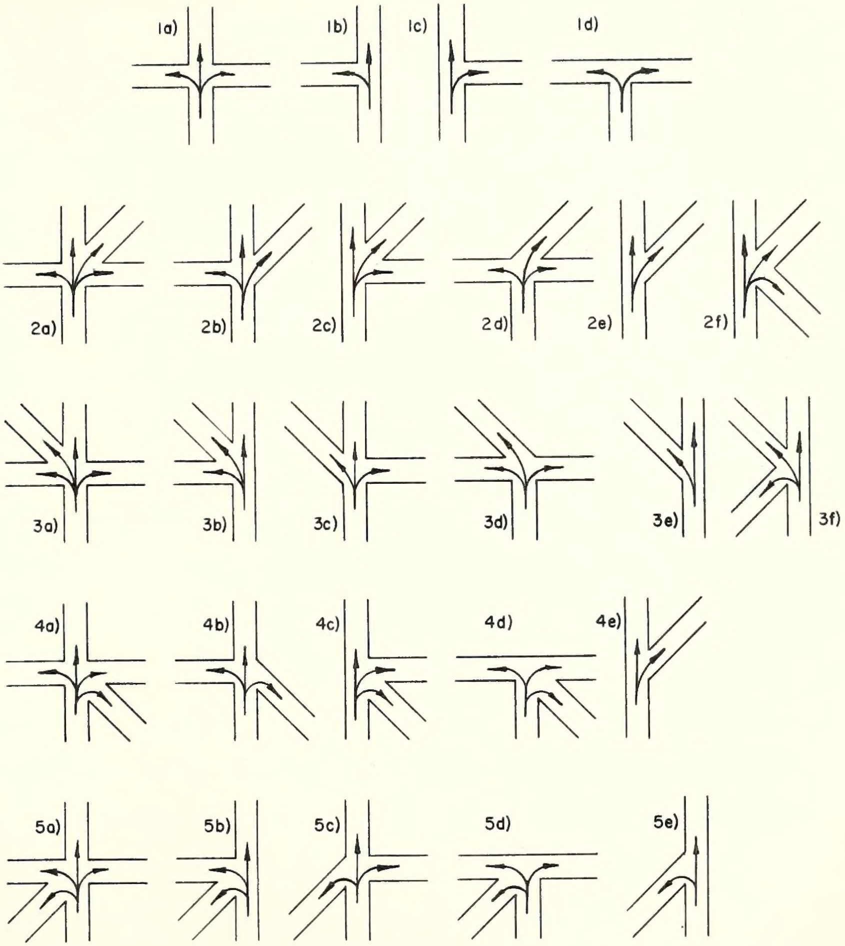


FIGURE A-1

INTERSECTION TYPES

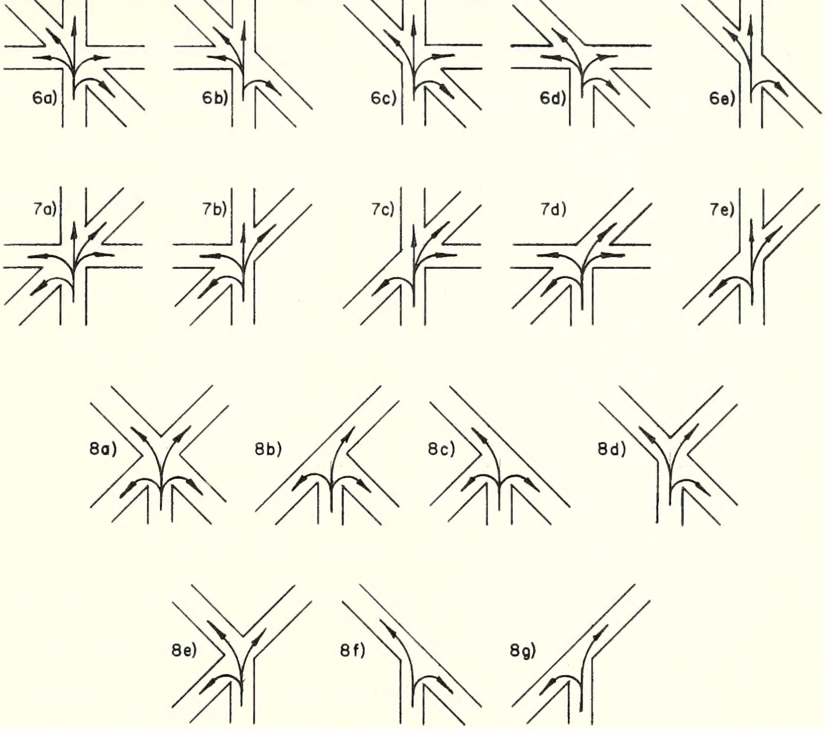


FIGURE A-1 , cont.

N_i = number of lanes in link i. N_{mzx} = 3.

$LLAN_i$ = length of each lane, in feet, from center of tail intersection to center of head intersection. Left and right turning pockets are measured from the center of the head intersection back to the beginning of the lane.

$CURBD_i$ = distance from center of head intersection back to stop line.

$IPARKR_i$ = parking on right side of link, yes or no.

$IPARKL_i$ = parking on left side of link, yes or no.

$IOPP_{i,j,j=1,5}$ = opposing links beginning with rightmost and proceding counter-clockwise. The directly opposite link (i.e. opposing left turns) is indexed by j = 3. Opposing links on the right are indexed by j = 1 and j = 2; if only one exists, it is indexed by j = 1. Opposing links on the left are indexed by j = 4 and j = 5; if only one exists, it is indexed by j = 5.

$NREC_{i,j,j=1,6}$ = receiving link for each of six possible turning movements, beginning with the rightmost and proceding counterclockwise. Straight-through movements in lane 1 and lane 2 are indexed by j = 3 and j = 4 respectively. One right turn is indexed by j = 1; two are indexed by j = 1 and j = 2. One left turn is indexed by j = 6; two are indexed by j = 5 and j = 6.

Turning radii are not required for all links, since a table of default radii is kept in PRESIG. These default values were developed from criteria found in the AASHO

Policy on Geometric Design of Rural Highways (1), and are shown in Table A-1. For those links with radii differing from the default values it is necessary to supply the proper values with the rest of the input data.

Revised discharge boundary values may be input prior to the final output similarly to the turning radii values. In both cases an input value of zero is assumed to mean no change is required. In the case of turning radii, the default value is used. In the case of discharge boundaries, the calculated value is used.

The PRESIG input deck consists of the cards given in Table A-2 in the specified order.

The PRESIG program code is given in Table A-3.

TABLE A-1

Default Radii Used in PRESIG

Angle of Turn[1]	Turn Type	Default Radius
135°	Right	27 ft.
90°	Right	30 ft.
45°	Right	70 ft.
45°	Free Left	70 ft.
45°	Delayed Left	50 ft.
90°	Free Left	40 ft.
90°	Delayed Left	30 ft.
135°	Free Left	27 ft.
135°	Delayed Left	27 ft.

---

[1] Angle of turn is measured from the extension of the centerline tangent of the approach link to the corresponding tangent on the receiving link.

TABLE A-2

PRESIG Punched Card Input Format

Card Designation	Card Description	Column	Symbol	Meaning	Units	Format
A	Program Parameters	1	A	Card Type	---	A
		6-7	NOLINK	No. of links in network	---	XX
		8-12	WIDTH	Lane width	ft.	XXXX.
		13-17	PARKW	Parking lane width	ft.	XXXX.
		18-21	NOUTPT	Number of output links	---	XXXX
B	Link Variables[a]	1	B	Card Type	---	B
		3-4	L	Link number	---	XX
		6	ITYPE	Intersection type, numerical	---	X
		7	JTYPE	Intersection type, alphabetical	---	X
		8	N	Number of lanes	---	X
		9-12	LLAN(1)	Lane length, lane 1	ft.	XXXX
		13-16	LLAN(2)	Lane length, lane 2	ft.	XXXX
		17-20	LLAN(3)	Lane length, lane 3	ft.	XXXX
		21-23	CURBD	Distance from center of head intersection back to stop line.	ft.	XXX

TABLE A-2, Cont.

Card Designation	Card Description	Column	Symbol	Meaning	Units	Format
		24-26	IPARKR	Parking on right (yes or no)	---	XXX[b]
		27-29	IPARKL	Parking on left (yes or no)	---	XXX[b]
		30-31	IOPP(1)	First opposing link number	---	XX
		⋮	⋮	⋮		
		38-39	IOPP(5)	Fifth opposing link number	---	XX
		40-41	LNKADJ	Adjacent link	---	XX
		42-43	NREC(1)	Receiving link, turn #1	---	XX
		⋮	⋮	⋮		
		52-53	NREC(6)	Receiving link, turn #6	---	XX
C	Radius Card[e]	1	C	Card type	---	C
		4-5	LHOLD	Link number	---	XX
		7-10	RAD(1)	Preset radius, turn number 1	ft.	XXX.[c]
		⋮	⋮	⋮		⋮
		27-30	RAD(6)	Preset radius, turn number 6	ft.	XXX.[c]

TABLE A-2, Contd.

Card Designation	Card Description	Column	Symbol	Meaning	Units	Format
D	Revised Discharge Boundary Card^{e}	1	D	Card type	---	D
		4-5	L	Link number	---	XX
		6-9	DNEW(1)	Revised Discharge Boundary for turn number 1	ft.	XXXX^{d}
		⋮	⋮	⋮		⋮
		26-29	DNEW(6)	Revised Discharge Boundary for turn number 6	ft.	XXX^{d}
E	Revised Begin-Turn Point Card^{e}	1	F	Card Type	---	F
		4-5	L	Link number	---	XX
		6-9	BGNNEW(1)	Revised Begin-Turn point for turn no. 1	---	XXXX^{d}
		⋮	⋮	⋮		⋮
		26-29	BGNNEW(6)	Revised Begin-Turn point for turn no. 6	---	XXXX^{d}

TABLE A-2, Cont.

[a] One card for each link.

[b] Word must be left-justified in the field.

[c] Blank field does not cause radius to be preset.

[d] Blank field causes no revision to that turn.

[e] One card required for each link being revised. Series must be concluded with an END (Col. 1-3) card. END card must be present even if no revisions are made.

TABLE A-3

PRESIG PROGRAM CODE

```
C************************************************************************
C************************************************************************
C
C
C                                 P R E S I G
C
C
C                          A PRELIMINARY GEOMETRY
C                  DETERMINATION PROGRAM FOR USE WITH
C
C                                 S I G N E T
C
C
C************************************************************************
C************************************************************************
C
C
C
      PROGRAM MAIN (INPUT,OUTPUT,PUNCH,TAPE5=INPUT,TAPE6=OUTPUT,
     1      TAPE7=PUNCH)
      DIMENSION ITYPE(99),JTYPE(99),IOPP(99,5),NMPRK(2),JPARK(99),
     1NOLANE(99),R(6),XLINK(99,3)
      COMMON/A/DB(99),CURBD(99),RAD(99,6),DBO(99,9),BEGINPT(99,6),
     1      STPLINE(99),DBNDRY(99,6),NREC(99,6)
      COMMON/B/FLPARK(99),PARKW,WIDTH,W(99)
      COMMON/C/LNKADJ(99)
      COMMON/D/FLNOLN(99)
      DIMENSION KSW(6,7,8)
      DIMENSION DNEW(99,6)
      DIMENSION FLPARKL(99),FN(99),BGNNEW(99,6)
      REAL LLAN(99,3)
      INTEGER TYPE
      DATA NMPRK/3HYES,3HNO /
      DATA R/30.,50.,50.,70.,30.,40./
C
C     R CONTAINS DEFAULT RADIUS VALUES
C
      DATA KSW/2,0,4,5,0,7,0,0,4,5,0,7,2,0,4,5,0,0,2,0,0,0,0,7,2,0,4,5,0
     1,7,12*0,2,3,4,5,0,7,3,0,4,5,0,7,2,3,4,5,0,0,2,3,0,0,0,7,3,0,4,5,0,
     30,1,3,4,5,0,0,6*0,2,0,4,5,6,7,0,0,4,5,6,7,2,0,4,5,0,6,2,0,0,0,6,7,
     40,0,4,5,0,6,0,0,4,5,6,8,6*0,1,2,4,5,0,7,1,0,4,5,0,7,1,2,4,5,0,0,1,
     52,0,0,0,7,1,0,4,5,0,0,12*0,2,0,4,5,7,8,0,0,4,5,7,8,2,0,4,5,0,8,2,0
     6,0,0,7,8,0,0,4,5,0,8,12*0,1,2,4,5,6,7,1,0,4,5,6,7,1,2,4,5,0,6,1,2,
     70,0,6,7,1,0,4,5,0,6,12*0,2,3,4,5,7,8,3,0,4,5,7,8,2,3,4,5,0,8,2,3,0
     8,0,7,8,3,0,4,5,0,8,12*0,1,3,0,0,6,8,1,3,0,0,0,8,1,0,0,0,  6,8,1,3,
     90,0,0,6,3,0,0,0,6,8,1,0,0,0,0,6,3,0,0,0,0,8/
C
      DATA IPNCH/1/
C
C     IPNCH=0 PRODUCES NO PUNCH OUTPUT
C     IPNCH≠0 PRODUCES PUNCH OUTPUT
C
      DO 90 L=1,99
      DO 85 I=1,9
   85 DBO(L,I)=0.
      DO 90 I=1,6
      RAD(L,I)=0.
   90 BEGINPT(L,I)=0.
      KNTCARD=1
C
```

TABLE A-3, cont

```
      READ (5,1) TYPE,NOLINK,WIDTH,PARKW,NOUTPT
    1 FORMAT (A1,4X,I2,F5.0,F5.0,I4)
C
C     PROGRAM PARAMETERS CARD
C
C     TYPE= CARD TYPE (A)
C     NOLINK= NO. OF LINKS IN NETWORK
C     WIDTH= NETWORK-WIDE LANE WIDTH
C     PARKW=NETWORK-WIDE PARKING LANE WIDTH
C     NOUTPT= NUMBER OF OUTPUT LINKS
C
      IF (TYPE.NE.1HA) CALL DIAGN(1,KNTCARD)
      WRITE (6,2) NOLINK,WIDTH,PARKW
    2 FORMAT (1H1,4X,125(1H*),/5X,125(1H*),///53X,*SYSTEM GEOMETRY DETER
     1MINATION*,///5X,125(1H*),/5X,125(1H*),/////20X,*NUMBER OF LINKS =
     2*,I4,//25X,*LANE WIDTH = *,F4.0,* FEET*,//17X,*PARKING LANE WIDTH
     3= *,F4.0,* FEET*,//////20X,*INT. TYPE*,5X,*LANES*,5X,*LANE LENGTH
     4(1-3)*,9X,*OPPOSING LINKS (1-5)*,14X,*PARKING*,5X,*ADJ. LINK*///)
      ISTART=NOUTPT+1
      DO 120 LINK=1,NOLINK
C
      READ (5,3) TYPE,L,ITYPE(LINK),JTYPE(LINK),N,(LLAN(LINK,I),I=1,3),
     1CURBD(LINK),IPARKR,IPARKL,(IOPP(LINK,I),I=1,5),LNKADJ(LINK),(NREC(
     2LINK,I),I=1,6)
    3 FORMAT (A1,1X,I2,1X,I1,A1,I1,3F4.0,F3.0,2A3,13I2)
C
C     LINK VARIABLES CARDS
C
C     TYPE= CARD TYPE (B)
C     L= LINK NUMBER
C     ITYPE= INTERSECTION TYPE, NUMERICAL
C     JTYPE= INTERSECTION TYPE, ALPHABETICAL
C     N= NUMBER OF LANES IN LINK
C     LLAN= LENGTH OF EACH LANE, CENTER OF INTERSECTION TO CENTER OF
C     INTERSECTION.  LEFT AND RIGHT TURN POCKETS ARE MEASURED FROM
C     CENTER OF INTERSECTION BACK TO BEGINNING OF LANE.
C     CURBD= DISTANCE FROM CENTER OF HEAD INTERSECTION BACK TO STOP LINE
C     IPARK= PARKING  (YES OR NO, LEFT JUSTIFIED IN FIELD)
C     IOPP= OPPOSING LINKS, BEGINNING WITH RIGHTMOST AND PROCEDING
C     COUNTERCLOCKWISE.
C     LNKADJ= ADJACENT LINK (PARALLEL,BUT OPPOSITE DIRECTION)
C     NREC= RECEIVING LINK FOR EACH OF 6 TURNS, BEGINNING WITH RIGHTMOST
C     AND PROCEDING COUNTERCLOCKWISE.  STRAIGHT MOVEMENTS, LANE 1
C     AND LANE 2, ARE ALWAYS NUMBERED 3 AND 4, RESPECTIVELY.  ONE
C     RIGHT TURN IS NUMBERED 1. TWO RIGHT TURNS ARE NUMBERED 1 AND
C     2. ONE LEFT TURN IS NUMBERED 6.  TWO LEFT TURNS ARE NUMBERED
C     5 AND 6.
C
      KNTCARD=KNTCARD+1
      IF (TYPE.NE.1HB) CALL DIAGN(2,KNTCARD)
      IF (L.NE.LINK) CALL DIAGN(3,KNTCARD)
      IF (N.GT.3) CALL DIAGN(4,KNTCARD)
      IF (N.GT.3) N=3
      IF (N.LT.1) CALL DIAGN(8,KNTCARD)
      IF (N.LT.1) N=1
      IF (IPARKR.EQ.NMPRK(2)) GO TO 100
      JPARK(LINK)=1
      GO TO 110
  100 JPARK(LINK)=0
  110 JP=JPARK(LINK)+1
      FLPARK(LINK)=FLOAT(JPARK(LINK))
```

```
      FLPARKL(LINK)=0.
      IF (IPARKL.EQ.NMPRK(1)) FLPARKL(LINK)=1.
      NOLANE(LINK)=N
      FLNOLN(LINK)=FLOAT(NOLANE(LINK))
      WRITE (6,4) L,ITYPE(L),JTYPE(L),NOLANE(L),(LLAN(L,I),I=1,3),(IOPP(
     1L,I),I=1,5),NMPRK(JP),LNKADJ(L)
    4 FORMAT (10X,*LINK*,I3,7X,I1,A1,10X,I2,6X,3(F4.0,3X),5X,5(I2,3X),11
     1X,A3,10X,I2)
      IF (JTYPE(LINK).EQ.1HA) JTYPE(LINK)=1
      IF (JTYPE(LINK).EQ.1HB) JTYPE(LINK)=2
      IF (JTYPE(LINK).EQ.1HC) JTYPE(LINK)=3
      IF (JTYPE(LINK).EQ.1HD) JTYPE(LINK)=4
      IF (JTYPE(LINK).EQ.1HE) JTYPE(LINK)=5
      IF (JTYPE(LINK).EQ.1HF) JTYPE(LINK)=6
      IF (JTYPE(LINK).EQ.1HG) JTYPE(LINK)=7
      IF (JTYPE(LINK).EQ.1HH) JTYPE(LINK)=8
  120 CONTINUE
C
  125 READ (5,5) TYPE,LHOLD,(RAD(LHOLD,I),I=1,6)
    5 FORMAT (A1,2X,I2,1X,6(F4.1))
C
C     RADIUS CARD
C
C     TYPE= CARD TYPE (C)
C     LHOLD= LINK ON WHICH PRESET RADIUS IS TO BE USED
C     RAD= RADIUS OF EACH OF SIX POSSIBLE TURNS. RAD=0. CAUSES NO PRESET.
C
C     SERIES MUST BE CONCLUDED WITH AN END (COL. 1-3) CARD
C
      IF (EOF,5) 126,127
  126 KNTCARD=KNTCARD+1
      CALL DIAGN(5,KNTCARD)
      GO TO 135
  127 KNTCARD=KNTCARD+1
      IF (TYPE.EQ.1HE) GO TO 135
  130 IF (TYPE.NE.1HC) CALL DIAGN(6,KNTCARD)
      GO TO 125
  135 DO 140 LINK=ISTART,NOLINK
      L=LNKADJ(LINK)
      IF (L.EQ.0) GO TO 139
      NOL=NOLANE(L)
      FN(L)=FLNOLN(L)
      IF (FN(L).EQ.1) GO TO 138
      IF (LLAN(L,NOL).LT.LLAN(L,NOL-1).OR.LLAN(L,1).LT.LLAN(L,2)) FN(L)=
     1FN(L)-1
  138 WADJ=WIDTH*FN(L)+PARKW*FLPARK(L)
      GO TO 140
  139 WADJ=FLPARKL(LINK)*PARKW
  140 W(LINK)=WADJ+WIDTH*FLNOLN(LINK)+PARKW*FLPARK(LINK)
      DO 1000 LINK=ISTART,NOLINK
      IT=ITYPE(LINK)
      JT=JTYPE(LINK)
      L=LNKADJ(LINK)
      IF (L.EQ.0) GO TO 141
      WADJ=WIDTH*FN(L)+PARKW*FLPARK(L)
      GO TO 142
  141 WADJ=FLPARKL(LINK)*PARKW
  142 CONTINUE
      GO TO (150,190,250,320,360,400,450,520) IT
  150 DO 155 I=1,6
  155 IF (RAD(LINK,I).EQ.0.) RAD(LINK,I)=R(I)
```

TABLE A-3, cont.

```
      GO TO (170,170,180,170,170) JT
  170 N=NREC(LINK,6)
      DBO(N,6)=DBOLT90(LINK,5,N)
      DBO(N,7)=DBOLT90(LINK,6,N)
      IF (JTYPE(LINK).EQ.2) GO TO 1000
  180 N=NREC(LINK,1)
      DBO(N,2)=DBORT90(LINK,1,N)
      GO TO 1000
  190 IF (JTYPE(LINK).EQ.2.OR.JTYPE(LINK).EQ.5.AND.RAD(LINK,1).EQ.0.)
     1RAD(LINK,1)=R(2)
      DO 195 I=1,6
  195 IF (RAD(LINK,I).EQ.0.) RAD(LINK,I)=R(I)
      GO TO (200,200,220,200,229,240) JT
  200 N=NREC(LINK,6)
      DBO(N,6)=DBOLT90(LINK,5,N)
      DBO(N,7)=DBOLT90(LINK,6,N)
      IF (JTYPE(LINK).EQ.2) GO TO 229
  220 N=NREC(LINK,1)
      DBO(N,2)=DBORT90(LINK,1,N)
  229 N=NREC(LINK,1)
      NT=1
      GO TO 232
  230 N=NREC(LINK,2)
      NT=2
  232 DBO(N,3)=DBORT2(LINK,NT,N)
      GO TO 1000
  240 N=NREC(LINK,1)
      IF (RAD(LINK,1).EQ.R(1)) RAD(LINK,1)=27.
      IF (RAD(LINK,2).EQ.R(2)) RAD(LINK,2)=R(1)
      DBO(N,1)=DBORT1(LINK,1,N)
      GO TO 230
  250 IF (JTYPE(LINK).NE.3.AND.JTYPE(LINK).NE.5) GO TO 251
      IF (RAD(LINK,5).EQ.0.) RAD(LINK,5)=R(3)
      IF (RAD(LINK,6).EQ.0.) RAD(LINK,6)=R(4)
      GO TO 254
  251 IF (JTYPE(LINK).NE.6) GO TO 254
      IF (RAD(LINK,5).EQ.0.) RAD(LINK,5)=25.
      IF (RAD(LINK,6).EQ.0.) RAD(LINK,6)=25.
  254 DO 255 I=1,6
  255 IF (RAD(LINK,I).EQ.0.) RAD(LINK,I)=R(I)
      GO TO (260,260,280,260,290,310) JT
  260 N=NREC(LINK,6)
      DBO(N,6)=DBOLT90(LINK,5,N)
      DBO(N,7)=DBOLT90(LINK,6,N)
      IF (JTYPE(LINK).EQ.2) GO TO 290
  280 N=NREC(LINK,1)
      DBO(N,2)=DBORT90(LINK,1,N)
  290 N=NREC(LINK,5)
      NT=3
      IF (JT.EQ.3.OR.JT.EQ.5) N=NREC(LINK,6)
      IF (JT.EQ.3.OR.JT.EQ.5) NT=5
      DBO(N,4)=DBOLT1(LINK,NT,N)
      NT=NT+1
      DBO(N,5)=DBOLT1(LINK,NT,N)
      GO TO 1000
  310 N=NREC(LINK,6)
      DBO(N,8)=DBOLT2(LINK,5,N)
      DBO(N,9)=DBOLT2(LINK,6,N)
      GO TO 290
  320 IF (RAD(LINK,1).EQ.0.) RAD(LINK,1)=27.
      IF (RAD(LINK,2).EQ.0.) RAD(LINK,2)=R(1)
```

TABLE A-3, cont.

```
      DO 325 I=3,6
  325 IF (RAD(LINK,I).EQ.0.) RAD(LINK,I)=R(I)
      GO TO (330,330,340,330,350) JT
  330 N=NREC(LINK,6)
      DBO(N,6)=DBOLT90(LINK,5,N)
      DBO(N,7)=DBOLT90(LINK,6,N)
      IF (JTYPE(LINK).EQ.2) GO TO 350
  340 N=NREC(LINK,2)
      DBO(N,2)=DBORT90(LINK,2,N)
  350 N=NREC(LINK,1)
      DBO(N,1)=DBORT1(LINK,1,N)
      GO TO 1000
  360 DO 365 I=1,2
  365 IF (RAD(LINK,I).EQ.0.) RAD(LINK,I)=R(I)
      IF (RAD(LINK,3).EQ.0.) RAD(LINK,3)=R(5)
      IF (RAD(LINK,4).EQ.0.) RAD(LINK,4)=R(6)
      IF (RAD(LINK,5).EQ.0.) RAD(LINK,5)=25.
      IF (RAD(LINK,6).EQ.0.) RAD(LINK,6)=25.
      GO TO (370,370,380,370,390) JT
  370 N=NREC(LINK,5)
      DBO(N,6)=DBOLT90(LINK,3,N)
      DBO(N,7)=DBOLT90(LINK,4,N)
      IF (JTYPE(LINK).EQ.2) GO TO 390
  380 N=NREC(LINK,1)
      DBO(N,2)=DBORT90(LINK,1,N)
  390 N=NREC(LINK,6)
      DBO(N,8)=DBOLT2(LINK,5,N)
      DBO(N,9)=DBOLT2(LINK,6,N)
      GO TO 1000
  400 IF (JTYPE(LINK).EQ.3.OR.JTYPE(LINK).EQ.5.AND.RAD(LINK,5).EQ.0.) RA
     1D(LINK,5)=R(3)
      IF (JTYPE(LINK).EQ.3.OR.JTYPE(LINK).EQ.5.AND.RAD(LINK,6).EQ.0.)
     1RAD(LINK,6)=R(4)
      IF (RAD(LINK,1).EQ.0.) RAD(LINK,1)=27.
      IF (RAD(LINK,2).EQ.0.) RAD(LINK,2)=R(1)
      DO 405 I=3,6
  405 IF (RAD(LINK,I).EQ.0.) RAD(LINK,I)=R(I)
      GO TO (410,410,420,410,430) JT
  410 N=NREC(LINK,6)
      DBO(N,6)=DBOLT90(LINK,5,N)
      DBO(N,7)=DBOLT90(LINK,6,N)
      IF (JTYPE(LINK).EQ.2) GO TO 430
  420 N=NREC(LINK,2)
      DBO(N,2)=DBORT90(LINK,2,N)
  430 N=NREC(LINK,1)
      DBO(N,1)=DBORT1(LINK,1,N)
      N=NREC(LINK,5)
      IF (JTYPE(LINK).EQ.3.OR.JTYPE(LINK).EQ.5) N=NREC(LINK,6)
      DBO(N,4)=DBOLT1(LINK,3,N)
      DBO(N,5)=DBOLT1(LINK,4,N)
      GO TO 1000
  450 IF (JTYPE(LINK).NE.2.AND.JTYPE(LINK).NE.6) GO TO 454
      IF (RAD(LINK,1).EQ.0.) RAD(LINK,1)=R(2)
  454 DO 459 I=1,6
      IF (I.GT.2) GO TO 455
      IF (RAD(LINK,I).EQ.0.) RAD(LINK,I)=R(I)
      GO TO 459
  455 IF (I.GT.4) GO TO 456
      IF (RAD(LINK,I).EQ.0.) RAD(LINK,I)=R(I+2)
      GO TO 459
  456 IF (RAD(LINK,5).EQ.0.) RAD(LINK,5)=25.
```

TABLE A 3, cont

```
      IF (RAD(LINK,6).EQ.0.) RAD(LINK,6)=25.
  459 CONTINUE
      GO TO (460,460,470,460,510) JT
  460 N=NREC(LINK,5)
      DBO(N,6)=DBOLT90(LINK,3,N)
      DBO(N,7)=DBOLT90(LINK,4,N)
      IF (JTYPE(LINK).EQ.2) GO TO 510
  470 N=NREC(LINK,1)
      DBO(N,2)=DBORT90(LINK,1,N)
  480 N=NREC(LINK,2)
      DBO(N,3)=DBORT2(LINK,2,N)
      GO TO 500
  500 N=NREC(LINK,6)
      DBO(N,8)=DBOLT2(LINK,5,N)
      DBO(N,9)=DBOLT2(LINK,6,N)
      GO TO 1000
  510 N=NREC(LINK,1)
      DBO(N,3)=DBORT2(LINK,1,N)
      GO TO 500
  520 IF (JTYPE(LINK).EQ.4.OR.JTYPE(LINK).EQ.6) GO TO 525
      IF (RAD(LINK,3).EQ.0.) RAD(LINK,3)=R(3)
      IF (RAD(LINK,4).EQ.0.) RAD(LINK,4)=R(4)
      IF (RAD(LINK,5).EQ.0.) RAD(LINK,5)=25.
      IF (RAD(LINK,6).EQ.0.) RAD(LINK,6)=25.
      GO TO 530
  525 IF (RAD(LINK,5).EQ.0.) RAD(LINK,5)=R(3)
      IF (RAD(LINK,6).EQ.0.) RAD(LINK,6)=R(4)
  530 IF (JTYPE(LINK).EQ.5.OR.JTYPE(LINK).EQ.7) GO TO 535
      IF (RAD(LINK,1).EQ.0.) RAD(LINK,1)=27.
      IF (RAD(LINK,2).EQ.0.) RAD(LINK,2)=R(2)
      GO TO 540
  535 IF (RAD(LINK,1).EQ.0.) RAD(LINK,1)=R(2)
  540 GO TO (550,560,550,551,560,551,560) JT
  550 N=NREC(LINK,5)
      NT=3
      GO TO 555
  551 N=NREC(LINK,6)
      NT=5
  555 DBO(N,4)=DBOLT1(LINK,NT,N)
      NT=NT+1
      DBO(N,5)=DBOLT1(LINK,NT,N)
      IF (JTYPE(LINK).EQ.4.OR.JTYPE(LINK).EQ.6) GO TO 570
  560 N=NREC(LINK,6)
      DBO(N,8)=DBOLT2(LINK,5,N)
      DBO(N,9)=DBOLT2(LINK,6,N)
  570 GO TO (580,580,580,580,590,580,590) JT
  580 N=NREC(LINK,1)
      DBO(N,1)=DBORT1(LINK,1,N)
      IF (JTYPE(LINK).EQ.3.OR.JTYPE(LINK).EQ.6) GO TO 1000
      N=NREC(LINK,2)
      GO TO 600
  590 N=NREC(LINK,1)
  600 DBO(N,3)=DBORT2(LINK,N,N)
 1000 CONTINUE
      DO 1105 L=1,NOLINK
      DB(L)=0.
      DO 1100 IDB=1,9
      IF (DBO(L,IDB).GT.DB(L)) DB(L)=DBO(L,IDB)
 1100 CONTINUE
 1105 CONTINUE
      DO 1210 L=ISTART,NOLINK
```

TABLE A-3, cont.

```
      NO=NCLANE(L)
      GO TO (1110,1120,1160) NO
1110  XLINK(L,1)=LLAN(L,1)-CURBD(L)-DB(L)
      XLINK(L,2)=0.
      XLINK(L,3)=0.
      GO TO 1200
1120  IF (LLAN(L,1)-LLAN(L,2)) 1130,1140,1150
1130  XLINK(L,1)=LLAN(L,1)-CURBD(L)
      XLINK(L,2)=LLAN(L,2)-CURBD(L)-DB(L)
1135  XLINK(L,3)=0.
      GO TO 1200
1140  XLINK(L,1)=LLAN(L,1)-CURBD(L)-DB(L)
      XLINK(L,2)=LLAN(L,2)-CURBD(L)-DB(L)
      GO TO 1135
1150  XLINK(L,1)=LLAN(L,1)-CURBD(L)-DB(L)
      XLINK(L,2)=LLAN(L,2)-CURBD(L)
      GO TO 1135
1160  IF (LLAN(L,1).GE.LLAN(L,2).AND.LLAN(L,1).GE.LLAN(L,3)) GO TO 1170
      IF (LLAN(L,2).GE.LLAN(L,3).AND.LLAN(L,2).GE.LLAN(L,1)) GO TO 1175
      GO TO1180
1170  XLINK(L,1)=LLAN(L,1)-CURBD(L)-DB(L)
      IF (LLAN(L,2).LT.LLAN(L,1)) GO TO 1173
      XLINK(L,2)=LLAN(L,2)-CURBD(L)-DB(L)
      IF (LLAN(L,3).EQ.LLAN(L,2)) GO TO 1172
1171  XLINK(L,3)=LLAN(L,3)-CURBD(L)
      GO TO 1200
1172  XLINK(L,3)=LLAN(L,3)-CURBD(L)-DB(L)
      GO TO 1200
1173  XLINK(L,2)=LLAN(L,2)-CURBD(L)
      IF (LLAN(L,3).EQ.LLAN(L,1)) GO TO 1172
      GO TO 1171
1175  XLINK(L,2)=LLAN(L,2)-CURBD(L)-DB(L)
      XLINK(L,1)=LLAN(L,1)-CURBD(L)
      IF (LLAN(L,3).EQ.LLAN(L,2)) GO TO 1172
      GO TO 1171
1180  XLINK(L,1)=LLAN(L,1)-CURBD(L)
      XLINK(L,2)=LLAN(L,2)-CURBD(L)
      GO TO 1172
1200  CONTINUE
      STPLINE(L)=AMAX1(XLINK(L,1),XLINK(L,2),XLINK(L,3))
1210  CONTINUE
      DO 1400 L=ISTART,NOLINK
      I=ITYPE(L)
      J=JTYPE(L)
      DO 1400 NT=1,6
      KGO=KSW(NT,J,I)
      IF (KGO.EQ.0) GO TO 1400
      GO TO (1300,1310,1320,1330,1330,1340,1350,1360) KGO
1300  CALL RTDIAG1 (L,NT)
      GO TO 1400
1310  CALL RT90 (L,NT)
      GO TO 1400
1320  CALL RTDIAG2 (L,NT)
      GO TO 1400
1330  CALL STRAITE (L,NT)
      GO TO 1400
1340  CALL LTDIAG1 (L,NT)
      GO TO 1400
1350  CALL LT90 (L,NT)
      GO TO 1400
1360  CALL LTDIAG2 (L,NT)
```

TABLE A-3, cont.

```
 1400 CONTINUE
      DO 1430 L=1,NOUTPT
      NOL=NOLANE
      DO 1410 LN=1,NOL
 1410 XLINK(L,LN)=400.
      STPLINE(L)=400.
      DO 1420 IT=1,6
 1420 DBNDRY(L,IT)=400.
 1430 CONTINUE
      DO 1439 I=1,NOLINK
      DO 1439 J=1,6
 1439 DNEW(I,J)=0.
C
 1440 READ(5,6) TYPE,L,(DNEW(L,I),I=1,6)
    6 FORMAT (A1,2X,I2,6F4.0)
C
C     REVISED DISCHARGE BOUNDARY CARD
C
C     TYPE= CARD TYPE (D)
C     L= LINK ON WHICH REVISED DISCHARGE BOUNDARY IS TO BE USED
C     DNEW=REVISED DISCHARGE BOUNDARY FOR EACH OF SIX POSSIBLE
C     TURNING MOVEMENTS.  DNEW=0 CAUSES NO REVISION.
C
C
C
      IF (EOF,5) 1441,1442
 1441 KNTCARD=KNTCARD+1
      CALL DIAGN(7,KNTCARD)
      GO TO 1449
 1442 KNTCARD=KNTCARD+1
      IF (TYPE.EQ.1HE) GO TO 1449
      IF (TYPE.NE.1HD) CALL DIAGN(8,KNTCARD)
      GO TO 1440
 1449 CONTINUE
      DO 1451 I=1,NOLINK
      DO 1451 J=1,6
 1451 IF (DNEW(I,J).NE.0.) DBNDRY(I,J)=DNEW(I,J)
      DO 1454 I=1,NOLINK
      DO 1454 J=1,6
 1454 BGNNEW(I,J)=0.
C
 1455 READ(5,15) TYPE,L,(BGNNEW(L,I),I=1,6)
   15 FORMAT (A1,2X,I2,6F4.0)
C
C      REVISED BEGIN-TURN POINT CARD
C
C     TYPE= CARD TYPE (F)
C     L= LINK ON WHICH REVISED BEGIN-TURN POINT IS TO BE USED
C     BGNNEW= REVISED BEGIN-TURN POINT FOR EACH OF SIX POSSIBLE TURN
C     TYPES.  BGNNEW=0 CAUSES NO REVISION.
C
C
      IF (EOF,5) 1456,1457
 1456 KNTCARD=KNTCARD+1
      CALL DIAGN (7,KNTCARD)
      GO TO 1459
 1457 KNTCARD=KNTCARD+1
      IF (TYPE.EQ.1HE) GO TO 1458
      IF (TYPE.NE.1HF) CALL DIAGN (8,KNTCARD)
      GO TO 1455
 1458 DO 1459 I=1,NOLINK
      DO 1459 J=1,6
```

TABLE A-3, cont.

```
1459 IF (RGNNEW(I,J).NE.0.) BEGINPT(I,J)=RGNNEW(I,J)
     WRITE (6,10)
  10 FORMAT (1H1,/////20X,*SUMMARY OF GEOMETRY INPUTS TO SIGNET*,/////2
    17X,*STOP*,14X,*LANE LENGTH*,24X,*DISCHARGE BOUNDARY*,/27X,*LINE*,
    212X,*1*,6X,*2*,6X,*3*,13X,*1*,6X,*2*,6X,*3*,6X,*4*,6X,*5*,6X,*6*.6
    3X//)
     DO 1450 L=1,NOLINK
1450 WRITE (6,11) L,STPLINE(L),(XLINK(L,I),I=1,3),(DBNDRY(L,I),I=1,6)
  11 FORMAT (10X,*LINK*,I3,10X,F4.0,10X,3(F4.0,3X),7X,6(F4.0,3X))
     WRITE (6,12)
  12 FORMAT (////////35X,*BEGINNING POINT OF TURN*,31X,*RADIUS OF TURN*
    1,/29X,*1*,6X,*2*,6X,*3*,6X,*4*,6X,*5*,6X,*6*,6X,7X,*1*,6X,*2*,6X,*
    23*,6X,*4*,6X,*5*,6X,*6*//)
     DO 1460 L=ISTART,NOLINK
1460 WRITE (6,13) L,(BEGINPT(L,I),I=1,6),(RAD(L,I),I=1,6)
  13 FORMAT (10X,*LINK*,I3,10X,6(F4.0,3X),7X,6(F4.0,3X))
     IF (IPNCH.EQ.0) STOP
     DO 1470 L=1,NOLINK
1470 WRITE (7,14) L,STPLINE(L),(DBNDRY(L,J),J=1,6),(XLINK(L,J),J=1,3),
    1(BEGINPT(L,J),J=1,6),(RAD(L,I),I=1,6)
  14 FORMAT (I2,3X,16F4.0,/5X,6F3.0)
     STOP
     END
```

```
     SUBROUTINE RTDIAG1 (L,NT)
     COMMON/A/DB(99),CURBD(99),RAD(99,6),DBO(99,9),BEGINPT(99,6),
    1  STPLINE(99),DBNDRY(99,6),NREC(99,6)
     N=NREC(L,NT)
     IF (DBO(N,1).LT.DB(N)) GO TO 100
     BEGINPT(L,NT)=STPLINE(L)+CURBD(L)-DB(N)
     DBNDRY(L,NT)=BEGINPT(L,NT)+.75*3.14159*RAD(L,NT)
     RETURN
 100 DX=DB(N)-DBO(N,1)
     BEGINPT(L,NT)=STPLINE(L)+CURBD(L)-DBO(N,1)
     DBNDRY(L,NT)=BEGINPT(L)+.75*3.14159*RAD(L,NT)+DX
     RETURN
     END
```

```
     SUBROUTINE RT90 (L,NT)
     COMMON/A/DB(99),CURBD(99),RAD(99,6),DBO(99,9),BEGINPT(99,6),
    1  STPLINE(99),DBNDRY(99,6),NREC(99,6)
     COMMON/B/FLPARK(99),PARKW,WIDTH,W(99)
     N=NREC(L,NT)
     DX=DB(N)-DBO(N,2)
     BEGINPT(L,NT)=STPLINE(L)+CURBD(L)-W(N)/2.+FLPARK(N)*PARKW+WIDTH/2.
    1-RAD(L,NT)
     DBNDRY(L,NT)=BEGINPT(L,NT)+.5*3.14159*RAD(L,NT)+DX
     RETURN
     END
```

```
     SUBROUTINE RTDIAG2 (L,NT)
     COMMON/A/DB(99),CURBD(99),RAD(99,6),DBO(99,9),BEGINPT(99,6),
    1  STPLINE(99),DBNDRY(99,6),NREC(99,6)
     COMMON/B/FLPARK(99),PARKW,WIDTH,W(99)
     N=NREC(L,NT)
```

TABLE A 3, cont.

```
      DX=DB(N)-DBO(N,3)
      BEGINPT(L,NT)=STPLINE(L)+CURBD(L)-RAD(L,NT)*TAN(3.14159/8.)-
     1 (W(N)/2.-WIDTH/2.-FLPARK(N)*PARKW)*TAN(3.14159/8.)
      DBNDRY(L,NT)=BEGINPT(L,NT)+.25*3.14159*RAD(L,NT)+DX
      RETURN
      END


      SUBROUTINE STRAITE (L,NT)
      COMMON/A/DB(99),CURBD(99),RAD(99,6),DBO(99,9),BEGINPT(99,6),
     1  STPLINE(99),DBNDRY(99,6),NREC(99,6)
      N=NREC(L,NT)
      DBNDRY(L,NT)=STPLINE(L)+CURBD(L)+DB(N)
      RETURN
      END


      SUBROUTINE LTDIAG1 (L,NT)
      COMMON/A/DB(99),CURBD(99),RAD(99,6),DBO(99,9),BEGINPT(99,6),
     1  STPLINE(99),DBNDRY(99,6),NREC(99,6)
      COMMON/B/FLPARK(99),PARKW,WIDTH,W(99)
      N=NREC(L,NT)
      NM=NT
      IF (NT.EQ.5) NT=3
      IF (NT.EQ.6) NT=5
      DXD=DB(N)-DBO(N,4)
      DXF=DB(N)-DBO(N,5)
      BEGINPT(L,NT)=STPLINE(L)+CURBD(L)-RAD(L,NT)*TAN(3.14159/8.)+
     1      1.414*(W(N)/2.-(FLPARK(N)*PARKW+WIDTH/2.))
      BEGINPT(L,NT+1)=STPLINE(L)+CURBD(L)-RAD(L,NT+1)*TAN(3.14159/8.)+
     11.414*(W(N)/2.-(FLPARK(N)*PARKW+WIDTH/2.))
      DBNDEL=BEGINPT(L,NT)+.25*3.14159*RAD(L,NT)+DXD
      DBNFREE=BEGINPT(L,NT+1)+.25*3.14159*RAD(L,NT+1)+DXF
      DBNDRY(L,NM)=AMAX1(DBNDEL,DBNFREE)
      NT=NM
      RETURN
      END


      SUBROUTINE LT90 (L,NT)
      COMMON/A/DB(99),CURBD(99),RAD(99,6),DBO(99,9),BEGINPT(99,6),
     1  STPLINE(99),DBNDRY(99,6),NREC(99,6)
      COMMON/B/FLPARK(99),PARKW,WIDTH,W(99)
      N=NREC(L,NT)
      NM=NT
      IF (NT.EQ.5) NT=3
      IF (NT.EQ.6) NT=5
      DXD=DB(N)-DBO(N,6)
      DXF=DB(N)-DBO(N,7)
      BEGINPT(L,NT)=STPLINE(L)+CURBD(L)-RAD(L,NT)+(W(N)/2.-(FLPARK(N)
     1*PARKW+WIDTH/2.))
      BEGINPT(L,NT+1)=STPLINE(L)+CURBD(L)-RAD(L,NT+1)+(W(N)/2.-(FLPARK(N
     1)*PARKW+WIDTH/2.))
      DBNDEL=BEGINPT(L,NT)+.5*3.14159*RAD(L,NT)+DXD
      DBNFREE=BEGINPT(L,NT+1)+.5*3.14159*RAD(L,NT+1)+DXF
      DBNDRY(L,NM)=AMAX1(DBNDEL,DBNFREE)
      NT=NM
      RETURN
      END
```

TABLE A-3, cont.

```
      SUBROUTINE LTDIAG2 (L,NT)
      COMMON/A/DB(99),CURBD(99),RAD(99,6),DBO(99,9),BEGINPT(99,6),
     1  STPLINE(99),DBNDRY(99,6),NREC(99,6)
      COMMON/B/FLPARK(99),PARKW,WIDTH,W(99)
      N=NREC(L,NT)
      NM=NT
      IF (NT.EQ.5) NT=3
      IF (NT.EQ.6) NT=5
      DXD=DB(N)-DBO(N,8)
      DXF=DB(N)-DBO(N,9)
      BEGINPT(L,NT)=STPLINE(L)+CURBD(L)-DBO(N,8)+1.414*(W(N)/2.-
     1(FLPARK(N)*PARKW+WIDTH/2.))
      BEGINPT(L,NT+1)=STPLINE(L)+CURBD(L)-DBO(N,9)+1.414*(W(N)/2.-(
     1FLPARK(N)*PARKW+WIDTH/2.))
      DBNDEL=BEGINPT(L,NT)+.75*3.14159*RAD(L,NT)+DXD
      DBNFREE=BEGINPT(L,NT+1)+.75*3.14159*RAD(L,NT+1)+DXF
      DBNDRY(L,NM)=AMAX1(DBNDEL,DBNFREE)
      NT=NM
      RETURN
      END
```

```
      REAL FUNCTION DBORT1(L,NT,N)
      COMMON/A/DB(99),CURBD(99),RAD(99,6),DBO(99,9),BEGINPT(99,6),
     1      STPLINE(99),DBNDRY(99,6),NREC(99,6)
      COMMON/B/FLPARK(99),PARKW,WIDTH,W(99)
      DBORT1=RAD(L,NT)/TAN(3.14159/8.)+(W(N)/2.-WIDTH/2.-PARKW*FLPARK(N)
     1)+1.414*(W(L)/2.-WIDTH/2.-PARKW*FLPARK(L))
      RETURN
      END
```

```
      REAL FUNCTION DBORT90(L,NT,N)
      COMMON/A/DB(99),CURBD(99),RAD(99,6),DBO(99,9),BEGINPT(99,6),
     1      STPLINE(99),DBNDRY(99,6),NREC(99,6)
      COMMON/B/FLPARK(99),PARKW,WIDTH,W(99)
      DBORT90=RAD(L,NT)+W(L)/2.-WIDTH/2.-FLPARK(L)*PARKW
      RETURN
      END
```

```
      REAL FUNCTION DBORT2(L,NT,N)
      COMMON/A/DB(99),CURBD(99),RAD(99,6),DBO(99,9),BEGINPT(99,6),
     1      STPLINE(99),DBNDRY(99,6),NREC(99,6)
      COMMON/B/FLPARK(99),PARKW,WIDTH,W(99)
      DBORT2=(RAD(L,NT)+W(N)/2.-WIDTH/2.-FLPARK(N)*PARKW)*TAN(3.14159/8.
     1)
      RETURN
      END
```

```
      REAL FUNCTION DBOLT1(L,NT,N)
      COMMON/A/DB(99),CURBD(99),RAD(99,6),DBO(99,9),BEGINPT(99,6),
     1      STPLINE(99),DBNDRY(99,6),NREC(99,6)
```

TABLE A-3, cont.

```
 COMMON/B/FLPARK(99),PARKW,WIDTH,W(99)
 DBOLT1=(RAD(L,NT)-W(N)/2.+WIDTH/2.+PARKW*FLPARK(N))*TAN(3.14189/8.
1)
 RETURN
 END



 REAL FUNCTION DBOLT90(L,NT,N)
 COMMON/A/DB(99),CURBD(99),RAD(99,6),DBO(99,9),BEGINPT(99,6),
1     STPLINE(99),DBNDRY(99,6),NREC(99,6)
 COMMON/B/FLPARK(99),PARKW,WIDTH,W(99)
 DBOLT90=RAD(L,NT)-WIDTH/2.
 RETURN
 END



 REAL FUNCTION DBOLT2(L,NT,N)
 COMMON/A/DB(99),CURBD(99),RAD(99,6),DBO(99,9),BEGINPT(99,6),
1     STPLINE(99),DBNDRY(99,6),NREC(99,6)
 COMMON/B/FLPARK(99),PARKW,WIDTH,W(99)
 COMMON/D/FLNOLN(99)
 DBOLT2=((RAD(L,NT)-W(N)+WIDTH+FLPARK(N)*PARKW)*1.414+RAD(L,NT)+WID
1TH*(FLNOLN(L)-.5)-W(L)/2.+PARKW*FLPARK(L))*1.414-RAD(L)+W(N)-WIDTH
2-PARKW*FLPARK(N)
 RETURN
 END



 SUBROUTINE DIAGN(N,NCARD)
 DIMENSION MESAG1(72),MESAG2(72)
 INTEGER FATAL(9)
 DATA (MESAG1(I),I=1,8)/80HINCORRECT CARD TYPE.  PROGRAM WAS EXPECT
1ING A TYPE-A CARD.                      /
 DATA (MESAG2(I),I=1,8)/80HERROR IS FATAL.  PROGRAM STOPPING.
1                                        /
 DATA (MESAG1(I),I=9,16)/80HINCORRECT CARD TYPE.  PROGRAM WAS EXPEC
1TING A TYPE-B CARD.                     /
 DATA (MESAG2(I),I=9,16)/80HERROR IS FATAL.  PROGRAM STOPPING.
1                                        /
 DATA (MESAG1(I),I=17,24)/80HINCORRECT LINK NUMBER ON CARD.
1                                        /
 DATA (MESAG2(I),I=17,24)/80HERROR IS NON-FATAL.  PROGRAM CONTINUIN
1G WITH POSSIBLE ERROR IN LINK DATA.     /
 DATA (MESAG1(I),I=25,32)/80HINCORRECT NUMBER OF LANES.  MAXIMUM NU
1MBER OF LANES PERMITTED IN SIGNET IS 3. /
 DATA (MESAG2(I),I=25,32)/80HERROR IS NON-FATAL.  PROGRAM CONTINUIN
1G WITH NUMBER OF LANES ASSUMED TO BE 3. /
 DATA (MESAG1(I),I=33,40)/80HFAILURE TO PLACE END CARD (COL. 1-3) A
1FTER TURNING RADIUS DATA.               /
 DATA (MESAG2(I),I=33,40)/80HERROR IS NON-FATAL.  PROGRAM CONTINUIN
1G.                                      /
 DATA (MESAG1(I),I=41,48)/80HINCORRECT CARD TYPE.  PROGRAM WAS EXPE
1CTING A TYPE-C CARD.                    /
 DATA (MESAG2(I),I=41,48)/80HERROR IS FATAL.  PROGRAM STOPPING.
1                                        /
 DATA (MESAG1(I),I=49,56)/80HFAILURE TO PLACE END CARD (COL. 1-3) A
1FTER DISCHARGE BOUNDARY DATA)           /
 DATA (MESAG2(I),I=49,56)/80HERROR IS NON-FATAL.  PROGRAM CONTINUIN
```

TABLE A-3, cont.

```
     1G.                                                                  /
      DATA (MESAG1(I),I=57,64)/80HINCORRECT CARD TYPE.  PROGRAM WAS EXPE
     1CTING A TYPE-D CARD.                                                /
      DATA (MESAG2(I),I=57,64)/80HERROR IS FATAL.  PROGRAM STOPPING.
     1                                                                    /
      DATA (MESAG1(I),I=65,72)/80HNUMBER OF LANES SET TO ZERO.  SIGNET A
     1SSUMES EACH LINK TO HAVE AT LEAST 1 LANE.                           /
      DATA (MESAG2(I),I=65,72)/80HERROR IS NON-FATAL.  PROGRAM CONTINUIN
     1G WITH NUMBER OF LANES ASSUMED TO BE 1.                             /
      DATA FATAL/1,1,2,2,2,1,2,1,2/
      WRITE (6,1) NCARD
    1 FORMAT (1H2,/5X,125(1H*),////20X,*ERROR ENCOUNTERED IN DATA CARD N
     1UMBER*,I4)
      INDEX=(N-1)*8+1
      IEND=INDEX+7
      WRITE (6,2) (MESAG1(I),I=INDEX,IEND),(MESAG2(I),I=INDEX,IEND)
    2 FORMAT (////30X,*ERROR IS *,8A10,///30X,8A10,////5X,125(1H*)///)
      IF (FATAL(N).EQ.2) GO TO 110
      RETURN
  110 STOP
      END
```

Appendix B

SIGNET Program Details

The following is a complete description of the inputs required by SIGNET. Input data for a simulatin run are recorded on code sheets, then punched on standard 80-column cards.

The SIGNET input deck consists of the following cards in the specified sequence, as indicated in Figure B-1:

Title Card	(1 card)
General Parameters Card	(1 card)
Program Instruction Cards	(6 cards)
Vehicle Length Card	(1 card)
Link Parameters Cards	(1 card per link, all links)
Link Geometry Cards	(2 cards per link, all links)
Signal Setting Cards	(2 cards per link, input and network links)
Turning Probability Cards	(1 card per link, repeated as desired, network and input links)
Input Volume Cards	(1 card per link, repeated as desired, input links)

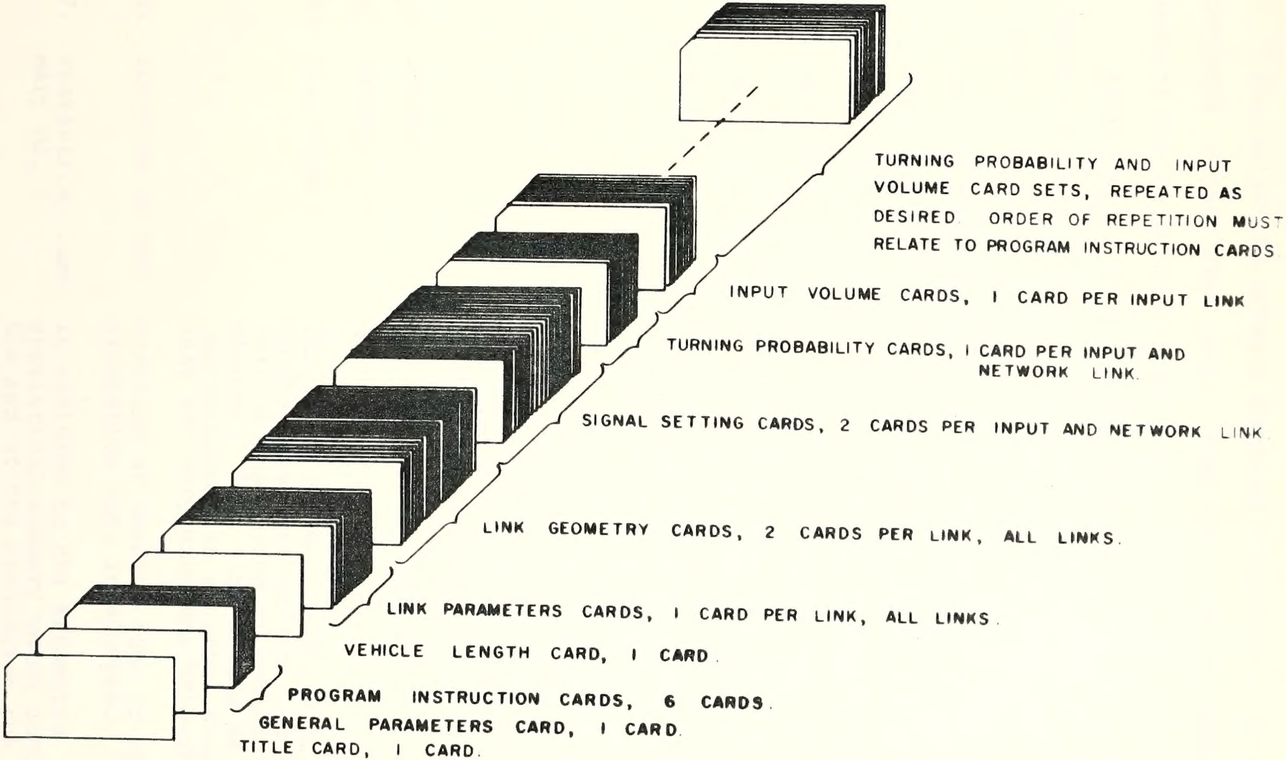

TURNING PROBABILITY AND INPUT VOLUME CARD SETS, REPEATED AS DESIRED. ORDER OF REPETITION MUST RELATE TO PROGRAM INSTRUCTION CARDS.
INPUT VOLUME CARDS, I CARD PER INPUT LINK
TURNING PROBABILITY CARDS, I CARD PER INPUT AND NETWORK LINK.
SIGNAL SETTING CARDS, 2 CARDS PER INPUT AND NETWORK LINK.
LINK GEOMETRY CARDS, 2 CARDS PER LINK, ALL LINKS.
LINK PARAMETERS CARDS, I CARD PER LINK, ALL LINKS.
VEHICLE LENGTH CARD, I CARD.
PROGRAM INSTRUCTION CARDS, 6 CARDS.
GENERAL PARAMETERS CARD, I CARD.
TITLE CARD, I CARD.

FIGURE B-I

INPUT DATA DECK STRUCTURE

The following gives a complete description of the contents of each card type. Specific variable names and formats may be found in Table B-1.

Title Card. This card allows the user to label the output with any alpha-numeric label up to 80 columns in length.

General Parameters Card. This card contains general information pertaining to the simulation run to be performed and the network size. The contents are described below:

Field Name	Field Contents
1. CYCLE	Amount of real-time simulated per simulation scanning cycle, in seconds. CYCLE should be between 1.0 and 5.0 seconds. A value of 2.0 seconds was used in this study.
2. NOLINK	Total number of links in the network. NOLINK must be less than 85. For larger networks program dimension statements must be changed.
3. NOUTPT	Number of output links.
4. NINPUT	Number of input links.
5. Fill Time	Duration of time simulated in order to fill the network with its initial population of vehicles, in minutes. Fill time should be approximately 15 minutes, although this may vary with the network.
6. Duration of Run	Duration of observe time in the simulation run, in minutes.
7. Statistical Summary No. 1	In addition to the automatic statistical summary which occurs at the end of the simulation run, up to 15 intermediate summaries may be quested. Field 7 contains the time,

Field Name	Field Contents
8. Statistical Summary No. 2 . . .	in minutes since the beginning of observe time, at which the first summary is to occur; Field 8, the second; etc. Any of these fields may be left blank, in which case the program assumes there are no further print options.
21. Statistical Summary No. 15	

Program Instruction Cards. This set of 6 cards contains instructions on both the proper times to perform several functions during the simulation run and the links to be considered in two extended bookkeeping operations. The contents are described below:

Field Name	Field Contents
Card No. 1:	
1. Turning Probability Revision Time 1 2. Turning Probability Revision Time 2 . . .	In addition to the initial turning probabilities read at the beginning of the simulation run, up to 10 revised turning probability sets may be employed. Field 1 contains the time, in minutes since the beginning of observe time, at which the first revision is to occur; Field 2, the second; etc. Any of these fields may be left blank, in which case the program assumes there are no more revisions to be made.
10. Turning Probability Revision Time 10	

Field Name	
Card No. 2:	
1. Input Volume Revision Time 1	In a manner similar to the turning probability revisions, up to 10 input volume revisions may be used. A blank field causes no more revisions to be made.
2. Input Volume Revision Time 2	
. . .	
10. Input Volume Revision Time 10	
Card No. 3:	
1. Re-initialization Time 1	As in the two cases above, a re-initialization of all bookkeeping variables may be requested. This option is used primarily in conjunction with the statistical summary option, causing a re-initialization immediately after each summary. Re-initialization after fill-time is automatic; therefore, it should not be requested here.
2. Re-initialization Time 2	
. . .	
10. Re-initialization Time 10	
Card No. 4:	
1. Begin Printout	Primarily for debugging purposes, a full printout of all vehicles' positions, velocities, accelerations, acceleration types, and turning movements may be specified. Field 1 contains the time, in seconds since the beginning of the simulation (i.e., beginning of fill time), that the printout is to begin. It is then made once each scan cycle until the time is greater than the -alue contained in Field 2. If both fields are blank no printout is made.
2. End Printout	

Field Name	Field Contents
3. Generator seed	To obtain a reproducable stream of random numbers it is necessary to provide a seed, or beginning number, for the pseudo-random number generator.

Card No. 5:

1. NTRACE	For more detailed analysis of travel times a route up to 15 links in length may be specified through the network. Field 1 contains the number of links in the route. Field 2 contains the first link in the route, Field 3 the second, etc. NTRACE must be greater than 0.
2. LKTRCE(1)	
3. LKTRCE(2)	
.	
.	
.	
16. LKTRCE(15)	

Card No. 6:

1. NQ	Similarly, a set of up to 10 links may be specified for queue length analysis. Field 1 contains the number of links in this set. Links are ordered within the set by link number. Field 2 contains the first link number, Field 3 the second, etc. NQ must be greater than 0.
2. LQ(1)	
3. LQ(2)	
.	
.	
.	
11. LQ(10)	

Vehicle Length Card. This card contains the parameters defining the effective length distribution for both cars and trucks. The contents are described below:

Field Name	Field Contents
1. $ELMN_1$ (Cars)	Mean effective length for cars, in feet. A value of 22.0 feet is reported in the literature for this variable.

Field Name	Field Contents
2. $ELSIG_1$ (Cars)	Effective Length standard deviation for cars.
3. $ELMN_2$ (Trucks)	Mean effective length for trucks, in feet.
4. $ELSIG_2$ (Trucks)	Effective length standard deviation for trucks.

Link Parameters Card. This card contains, for a specific link, lane relationships, link numbering relationships, and the parameters defining the target velocity distributions for cars and trucks on that link. One card of this type must be provided for each link. The contents are described below:

Field Name	Field Contents
1. i	Link number (identification). Links must be numbered in succession.
2. $NOLANE_i$	Width of link i in number of moving lanes at the intersection approach. $NOLANE_i$ = 1, 2, or 3.
3. $LANTN_{i,1}$	Lane required for turn. $LANTN_{i,k}$ = lane vehicle must be in to make turn number k.
4. $LANTN_{i,2}$	Turn numbers are discussed on page 55.
⋮	
8. $LANTN_{i,6}$	
9. $NREC_{i,1}$	Receiving link. $NREC_{i,k}$ = link which receives traffic from turning movement number k.
10. $NREC_{i,2}$	
⋮	
14. $NREC_{i,6}$	

Field Name	Field Contents
15. $\mathrm{LREC}_{i,1}$	Receiving Lane. $\mathrm{LREC}_{i,k}$ = lane which receives traffic from turning movement number k.
. . .	
20. $\mathrm{LREC}_{i,6}$	
21. $\mathrm{IOPP}_{i,1}$	Opposing link. Numbering conventions described on page 148.
22. $\mathrm{IOPP}_{i,2}$	
. . .	
25. $\mathrm{IOPP}_{i,5}$	
26. $\mathrm{TVELMN}_{i,cars}$	Mean target velocity for cars, in miles per hour.
27. $\mathrm{TVELSIG}_{i,cars}$	Target velocity standard deviation for cars.
28. $\mathrm{TVELMN}_{i,trucks}$	Mean target velocity for trucks, in miles per hour.
29. $\mathrm{TVELSIG}_{i,\ trucks}$	Target velocity standard deviation for trucks.

Link Geometry Cards. This pair of cards contains, for a specified link, all information related to the link geometry. The entire set of cards may be obtained as punched output from PRESIG, the companion geometry determination program, although this is not necessary. One pair of cards is required for each link. The contents are described below:

Field Name	Field Contents
Card No. 1:	
1. i	Link number (identification). Links must be numbered in succession.
2. $STPLINE_i$	Coordinate of the stopline, in feet.
3. $DBNDRY_{i,1}$	$DBNDRY_{i,k}$ = discharge boundary for turning movement number k.
4. $DBNDRY_{i,2}$	
⋮	
8. $DBNDRY_{i,6}$	
9. $XLINK_{i,1}$	$XLINK_{i,k}$ = length of lane k. (k = 1 = rightmost lane)
10. $XLINK_{i,2}$	
11. $XLINK_{i,3}$	
12. $BEGINPT_{i,1}$	$BEGINPT_{i,k}$ = beginning point of turn number k.
13. $BEGINPT_{i,2}$	
⋮	
17. $BEGINPT_{i,6}$	
Card No. 2:	
18. $RADIUS_{i,1}$	$RADIUS_{i,k}$ = radius of turn number k.
19. $RADIUS_{i,2}$	
⋮	
23. $RADIUS_{i,6}$	

Signal Setting Cards. This pair of cards contains, for a specified link, the fixed-time traffic signal settings. One pair must be provided for each input link and each networl link. The contents are described below:

Field Name	Field Contents
Card No. 1:	
1. i	Link number (identification). Links must be numbered in succession.
2. $LCYCLE_i$	Cycle length, in seconds.
3. $NOPHAS_{i,1}$	$NOPHAS_{i,k}$ = number of indications to lane k.
4. $NOPHAS_{i,2}$	$NOPHAS_{i,k}$ must be less than 6.
5. $NOPHAS_{i,3}$	
6. $NPHASE_{i,1,1}$	$NPHASE_{i,j,k}$ = number of k^{th} indication to lane j.
7. $NPHASE_{i,1,2}$	Indication numbers are defined in Table 5.
⋮	
10. $NPHASE_{i,1,5}$	
11. $NPHASE_{i,2,1}$	
12. $NPHASE_{i,2,2}$	
⋮	
20. $NPHASE_{i,3,5}$	

Field Name	Field Contents
Card No. 2:	
21. $LPHASE_{i,1,1}$	$LPHASE_{i,j,k}$ = duration of k^{th} indication to lane j, in seconds.
22. $LPHASE_{i,1,2}$	
$\vdots$	
25. $LPHASE_{i,1,5}$	
26. $LPHASE_{i,2,1}$	
$\vdots$	
35. $LPHASE_{i,3,5}$	
36. $LOFFSET_{i,1}$	$LOFFST_{i,k}$ = offset to beginning of first indication on lane k, in seconds.
37. $LOFFST_{i,2}$	
38. $LOFFST_{i,3}$	

Turning Probability Card. This card contains, for a specified link, the cumulative turning probabilities for the 6 possible turning movements. One card must be provided for each network and input link. The contents are described below:

Field Name	Field Contents
1. i	Link number. Links must be numbered in succession.
2. $PTURN_{i,1}$	$PTURN_{i,1}$ = probability that a vehicle leaving link i will use turning movement number 1.

Field Name	Field Contents
3. $PTURN_{i,2}$	$PTURN_{i,k} - PTURN_{i,k-1}$ = probability that a vehicle leaving link i will use turning movement Number k.
⋮	
7. $PTURN_{i,6}$	PTURN is a cumulative distribution.

Input Volume Card. This card contains, for a specified link, the hourly input volume and the percent trucks. One card must be provided for each input link. The contents are described below:

Field Name	Field Contents
1. i	Link number. Links must be numbered in succession.
2. VOL_i	Input Volume, vehicles per hour.
3. $PRTRUCK_i$	Percent trucks.

The use of successive sets of Turning Probability Cards and Input Volume Cards provides for modification of certain link characteristics while the simulation is in progress: viz., input volumes, input truck percentages, and turning probabilities. Care should be taken that these sets are in the correct order, as specified by the Program Instruction Cards. If revised turning probabilities and input volumes are required simultaneously, the turning probability cards should appear first. No items except those appearing in these two sets can be changed while a simulation run is in progress.

TABLE B-1

SIGNET Punched Card Input Format

Card Description	Card No.	Column	Symbol	Meaning	Units	FORTRAN Format	Notation
Title Card	-	1-80		Any message	---	8A10	----
General Parameters	-	1-2	CYCLE	Scan cycle	sec.	F2.0	X.
		3-5	NOLINK	No. of Links in network	---	I3	XXX
		6-8	NOUTPT	No. of Output Links	---	I3	XXX
		9-11	NINPUT	No. of Input Links	---	I3	XXX
		12-15	MINITIM	Fill Time	min.	I4	XXXX
		16-20	MLRUN	Observe Time	min.	I5	XXXXX
		21-24	SUMTIM(1)	First Summary Time	min.	I4	XXXX
		25-28	SUMTIM(2)	2nd Summary Time	min.	I4	XXXX
		.	.		.	.	.
		.	.	.	.	.	.
		.	.	.	.	.	.
		.	.	.	.	.	.
		77-80	SUMTIM(15)	15th Summary	min.	I4	XXXX
Program Instructions	1	1-4	NEWPR(1)	1st turning prob. revision	min.	I4	XXXX
		5-8	NEWPR(2)	2nd turning prob. revision	min.	I4	XXXX
		.	.		.	.	.
		.	.	.	.	.	.
		.	.	.	.	.	.
		.	.	.	.	.	.

TABLE B-1, Cont.

Card Description	Card No.	Column	Symbol	Meaning	Units	FORTRAN Format	Notation
		37-40	NEWPR(10)	10th turning prob. revision	min.	I4	XXXX
	2	1-4	NEWVOL(1)	1st input vol. revsion	min.	I4	XXXX
		5-8	NEWVOL(2)	2nd input vol. revision	min.	I4	XXXX
		.	.		.	.	.
		.	.	.	.	.	.
		.	.	.	.	.	.
		.	.	.	.	.	.
		37-40	NEWVOL(10)	10th input vol. revision	min.	I4	XXXX
Program	3	1-4	NEWINIT(L)	1st reinitialization	min.	I4	XXXX
		5-8	NEWINIT(2)	2nd reinitialization	min.	I4	XXXX
		.	.		.	.	.
		.	.	.	.	.	.
		.	.	.	.	.	.
		.	.	.	.	.	.
		37-40	NEWINIT(10)	10th reinitialization	min.	I4	XXXX
	4	1-5	IDEFST	Start printout	sec.	I5	XXXXX
		6-10	IDEFEND	End printout	sec.	I5	XXXXX
		11-30	ISEED	Random number seed	---	020	20 digit (octal)

TABLE B-1, Cont.

Card Description	Card No.	Column	Symbol	Meaning	Units	FORTRAN Format	Notation
	5	1-2	NTRACE	No. links in route	---	I2	XX
		3-5	LKTRCE(1)	1st link in route	---	I3	XXX
		6-8	LKTRCE(2)	2nd link in route	---	I3	XXX
		. . . .	. . . .	. . .	. . . .	. . . .	. . . .
		45-47	LKTRCE(15)	15th link in route	---	I3	XXX
	6	1-2	NQ	No. of links to be analyzed for queue length	---	I2	XX
		3-5	LQ(1)	1st link in set	---	I3	XXX
		6-9	LQ(2)	2nd link in set	---	I3	XXX
		. . .	. . .	. . .	. . .	. . .	. . .
		30-32	LQ(10)	10th link in set	---	I3	XXX
Vehicle Lengths		1-5	ELMN(cars)	Mean EL, cars	ft.	F5.2	XX.XX
		6-10	ELSIG(cars)	EL std. dev., cars	ft.	F5.2	XX.XX
		11-15	ELMN(trucks)	Mean EL, trucks	ft.	F5.2	XX.XX
		16-20	ELSIG	EL std. dev., trucks	ft.	F5.2	XX.XX

TABLE B-1, Cont.

Card Description	Card No.	Column	S-mbol	Meaning	Units	FORTRAN Format	Notation
Link Parameters	-	6	NOLANE	No. of Lanes	---	I1	X
		7	LANTN(1)	Lane req. for turn 1	---	I1	X
		8	LANTN(2)	Lane req. for turn 2	---	I1	X
		⋮	⋮	⋮	⋮	⋮	⋮
		12	LANTN(6)	Lane req. for turn 6	---	I1	X
		13-15	NREC(1)	Receiving link	---	I3	XXX
		16-18	NREC(2)	Receiving link	---	I3	XXX
		⋮	⋮	⋮	⋮	⋮	⋮
		28-30	NREC(6)	Receiving link	---	I3	XXX
		31-33	LREC(1)	Receiving link	---	I3	XXX
		⋮	⋮	⋮	⋮	⋮	⋮
		46-48	LREC(6)	Receiving link	---	I3	XXX
		49-51	IOPP(1)	Opposing link	---	I3	XXX
		⋮	⋮	⋮	⋮	⋮	⋮
		61-63	IOPP(5)	Opposing link	---	I3	XXX
		64-66	TVELMN(cars)	Mean target vel. (cars)	m.p.h.	F3.0	XX.

TABLE B-1, Cont.

Card Description	Card No.	Column	Symbol	Meaning	Units	FORTRAN Format	Notation
		70-72	TVELMN (trucks)	Mean target vel. (trucks)	m.p.h.	F3.0	XX.
		73-75	TVELSIG (trucks)	Std. dev. of targ. vel., trucks	m.p.h.	F3.0	XX.
Link Geometry	1	6-9	STPLINE	Stop-line	ft.	F4.0	XXXX
		10-13	DBNDRY(1)	Discharge Boundary	ft.	F4.0	XXXX
		.	.		.	.	.
		.	.	.	.	.	.
		.	.	.	.	.	.
		.	.	.	.	.	.
		30-33	DBNDRY(6)	Discharge Boundary	ft.	F4.0	XXXX
		37-40	XLINK(1)	Length of lane 1	ft.	F4.0	XXXX
		38-41	XLINK(2)	Length of lane 2	ft.	F4.0	XXXX
		42-45	XLINK(3)	Length of lane 3	ft.	F4.0	XXXX
		46-49	BEGINPT(1)	Begin-turn point	ft.	Fr.0	XXXX
		.	.		.	.	.
		.	.	.	.	.	.
		.	.	.	.	.	.
		66-69	BEGINPT(6)	Begin-turn point	ft.	F4.0	XXXX

TABLE B-1, Cont.

Card Description	Card No.	Column	Symbol	Meaning	Units	FORTRAN Format	Notation
Link Geometry (cont.)	2	6-8	RADIUS(1)	Radius of turn 1	ft.	F3.0	XXX
		⋮	⋮	⋮	⋮	⋮	⋮
		21-23	RADIUS(6)	Radius of turn 6	ft.	F3.0	XXX
Signal Settings	1	6-8	LCYCLE	Cycle length	sec.	I3	XXX
		9	NOPHAS(1)	No. of phases lane 1	---	I1	X
		10	NOPHAS(2)	No. of phases lane 2	---	I1	X
		11	NOPHAS(3)	No. of phases lane 3	---	I1	X
		12-13	NPHASE(1,1)	Phase No. 1, lane 1	---	I2	XX
		14-15	NPHASE(1,2)	Phase No. 2, Lane 1	---	I2	XX
		⋮	⋮	⋮	⋮	⋮	⋮
		20-21	NPHASE(1,5)	Phase No. 5, lane 1	---	I2	XX
		22-23	NPHASE(2,1)	Phase No. 1, lane 2	---	I2	XX
		⋮	⋮	⋮	⋮	⋮	⋮

TABLE B-1, Cont.

Card Description	Card No.	Column	Symbol	Meaning	Units	FORTRAN Format	Notation
		40-41	NPHASE(3,5)	Phase No. 5, lane 3	---	I2	XX
	2	6-8	LPHASE(1,1)	Duration #1, lane 1	sec.	I3	XXX
		.	.		.	.	.
		.	.	.	.	.	.
		.	.	.	.	.	.
		18-20	LPHASE(1,5)	Duration #5, lane 1	sec.	I3	XXX
		21-23	LPHASE(2,1)	Duration #1, lane 2	sec.	I3	XXX
		.	.		.	.	.
		.	.	.	.	.	.
		.	.	.	.	.	.
		.	.	.	.	.	.
		48-50	LPHASE(3,5)	Duration #5, lane 3	sec.	I3	XXX
		51-53	LOFFST(1)	Offset, lane 1	sec.	I3	XXX
		54-56	LOFFST(2)	Offset, lane 2	sec.	I3	XXX
		57-59	LOFFST(3)	Offset, lane 3	sec.	I3	XXX
Turning Probability	-	6-9	PTURN(1)	Turning Probability, turn 1	---	F4.2	X.XX
		10-13	PTURN(2)	Turning Probability, turn 2 (cumulative)	---	F4.2	X.XX
		.	.		.	.	.
		.	.		.	.	.
		.	.		.	.	.
		.	.	.	.	.	.
		.	.	.	.	.	.
		.	.	.	.	.	.

TABLE B-1, Cont.

Card Description	Card No.	Column	Symbol	Meaning	Units	FORTRAN Format	Notation
		26-29	PTURN(6)	Turning Probability, turn 6 (cumulative)	---	F4.2	X.XX
Input Volume	-	6-11	VOL	Volume	v.p.h.	F6.0	XXXXXX
		12-15	PRTRUCK	Percent trucks	---	F4.2	X.XX

A complete listing of the SIGNET program code is given in Table B-2. The program is written entirely in FORTRAN IV. Approximately 121000_8 central memory words are required to run the program at its current 84-link, 1500 vehicle capacity; the size of the program may be reduced to match the requirements of the application by reducing both the link capacity and the vehicle capacity. The means of doing this is explained below.

Several routines referenced by SIGNET are system routines on the CDC 6500. It is probable that the program could not be run on other installations without provision being made for supplying at least some of these routines or their equivalent. They are as follows:

EXP(X): A real function for finding e^x.

SQRT(X): A real function used for finding X.

ALOG(X): A real function used for finding ln(x).

RANF(A): A real function used to generate a sequence of pseudo-random numbers which are uniformly distributed in (0,1). In order to generate a sequence of random numbers used RANF with a zero argument:

```
Y = RANF(0).
```

In order to restart the sequence at some given point, use RANF with the argument set to some positive seed, written in octal notation:

```
X = 17162534251617262534B;
Y = RANF(X);
```

then use RANF(0).

The random number is determined from:

$$X_{i+1} = K * X_i \text{ MOD } 2^{48}$$

where

X_{i+1} = the new random number.

X_0 = the seed; and

K = 20000000000000055364B.

ISHFTR(X,M): Causes word X to be shifted right M bits. This COMPASS function is used in unpacking the ICAR word.

ISHFTL(X,M): Causes word X to be shifted left M bits. This COMPASS function is used in packing the ICAR word.

The CDC 6500 employs a 60-bit word. For installations employing a smaller word revisions would be necessary to the functions IBITM and JBITM (used for unpacking and packing the ICAR word, respectively).

Storage for vehicles is allocated in a way that minimizes the amount of wasted space. In its present state the program will accommodate up to 1500 vehicles at any time. Reducing this capacity for applications with potentially fewer vehicles will result in reduced field length and greater economy. The reduction is easily accomplished by changing the dimension of the variables POSN, VEL, ACCEL, ICAR, NXTCAR, and IPRECAR in common block A1 and the variable IHOLE in common block ZQ. The dimension id equal to the vehicular capacity of the program.

If the program capacity is exceeded at any time a message is printed to that affect and the program continues. As a guide in determining what the program capacity should be, the maximum number of vehicles in the network is printed at the end of each run.

The SIGNET program deck structure is shown in Figure B-2.

TABLE B-2

SIGNET PROGRAM CODE

```
C     ******************************************************************
C     ******************************************************************
C
C
C           SIGNET-------A PROGRAM FOR THE SIMULATION OF A
C                                TRAFFIC SIGNAL NETWORK
C
C
C     ******************************************************************
C     ******************************************************************
C
C
      PROGRAM SIGNET (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
      COMMON /A1/ IREM,INEXT,IMAX,ISIZE,IFRSTCR(84,3),ILSTCAR(84,3),VEL(
     11500),POSN(1500),ACCEL(1500),ICAR(1500),NXTCAR(1500),IPRECAR(1500)
      COMMON /Z0/ IHOLE(1500),LASTONE
      COMMON /PARAM/ A0,FREEK,REACT,CYCLE
      COMMON /LINKS/ LANTN(84,6),XLINK(84,3)
      COMMON /SIG/ LCYCLE(84),NOPHAS(84,3),NPHASE(84,3,5),LPHASE(84,3,5)
     1,LOFFST(84,3)
      COMMON /RELAT/ LINCAP(84),NOLINK,LANCAP(84,3),NOLANE(84)
      COMMON /GEOMET/ RADIUS(84,6),BEGINPT(84,6)
      COMMON /OPP/ ICPP(84,5)
      COMMON /MSK/ MASK(20)
      COMMON /GEOM/ STPLINE(84),RANDEV(84,6)
      COMMON /TURN/ NREC(84,6),LREC(84,6),PTURN(84,5),PTR(84,2)
      COMMON /B2/ TTIME(84),TDIST(84),IRTIME(15,105),TTTIME,TTDIST,HSTO(
     115,3,14)
      COMMON /BUK/ NHOLD(84,3)
      COMMON /QU/ LQUEUE(15,3),LQ(10),AQ
      COMMON /PV/ P(84,3),V(84,3)
      COMMON /TM/ ITIME
      COMMON /TRCE/ NTRACE,LKTRCE(15),NOTRC(15),INTTRC,KNTRAC(15)
      COMMON /KMNTZ/ NCAR(84),NOUTPT,ISTART,IEND,KGEN(84)
      COMMON /TIMKK/ INITIM,LRUN,MINITIM
      COMMON /CAPACTY/ LNCAP
      COMMON /PARAM2/ TVELMN(84,2),TVELSIG(84,2),ACCMN(2),ACCSIG(2),DECM
     1N(2),DECSIG(2),FLMN(2),FLSIG(2),PRTRUCK(84)
      DIMENSION MESAGE(8)
      DIMENSION NEWPR(10), NEWVOL(10)
      DIMENSION NEWINIT(10)
      INTEGER SUMTIM(15),SUMKNT
      REAL XLINK,LFREE,LDELAY
      DATA LNCAP/30/
      DATA ISIZE/1500/
      DATA A0,FREEK,REACT,ACCMN,ACCSIG,DECMN,DECSIG/30.,.10,.75,8.,4.,1.
     1,.5,4.,4.,.5,.5/
      MASK(1)=1B
      MASK(2)=3B
      MASK(3)=7B
      MASK(4)=17B
      MASK(5)=37B
      MASK(6)=77B
      MASK(7)=177B
      MASK(9)=777B
      MASK(12)=7777B
      MASK(13)=17777B
      TDIST=0.
      TTIME=0.
C
```

TABLE B-2, cont.

```
C     ******************************************************************
C
C          READ PROGRAM DATA
C
C     ******************************************************************
C
      READ (5,320) (MESAGE(I),I=1,8)
      READ (5,330) CYCLE,NOLINK,NOUTPT,NINPUT,MINITIM,MLRUN,(SUMTIM(I),I
     1=1,15)
      INITIM=MINITIM*60
      LRUN=MLRUN*60
      DO 100 I=1,6
      IF (SUMTIM(I).EQ.0) GO TO 100
      SUMTIM(I)=SUMTIM(I)*60+INITIM
  100 CONTINUE
      READ (5,340) (NEWBR2(I),I=1,10)
      READ (5,340) (NEWVOL(I),I=1,10)
      READ (5,340) (NEWINIT(I),I=1,10)
      READ (5,350) IOFFST,IOFFEND,ISEED
      RN=RANF(ISEED)
      READ (5,360) NTRACE,(LKTRCE(N),N=1,NTRACE)
      READ (5,370) NO,(LO(N),N=1,NO)
      READ (5,380) (FLMN(I),FLSIG(I),I=1,2)
      DO 110 L=1,NOLINK
      READ (5,390) NOLANE(L),(LANTN(L,J),J=1,6),(NREC(L,J),J=1,5),(LREC(
     1L,J),J=1,6),(IOPP(L,J),J=1,5),(TVFLMN(L,J),TVFLSIG(L,J),J=1,2)
  110 CONTINUE
C
C     IOPP(3) IS THE OPPOSING LINK WHICH HAS GREEN SIMULTANEOUSLY WITH
C     THE LINK UNDER CONSIDERATION.  IOPP(1) AND IOPP(2) ARE OPPOSING
C     LINKS TO THE RIGHT.  IOPP(4) AND IOPP(5) ARE OPPOSING LINKS TO THE
C     LEFT.
C
      DO 120 L=1,NOLINK
  120 READ (5,400) STPLINE(L),(BONDRY(L,J),J=1,6),(XLINK(L,J),J=1,3),(BE
     1GINPT(L,J),J=1,6),(RADIUS(L,J),J=1,5)
      NO=NOUTPT+1
      DO 130 L=NO,NOLINK
  130 READ (5,410) LCYCLE(L),(NOPHAS(L,J),J=1,3),((NPHASE(L,J,K),K=1,5),
     1J=1,3),((LPHASE(L,J,K),K=1,5),J=1,3),(LOFFST(L,J),J=1,3)
      DO 140 L=1,NOUTPT
      LCYCLE(L)=60
      NL=NOLANE(L)
      DO 140 LN=1,NL
      NOPHAS(L,LN)=1
      NPHASE(L,LN,1)=2
      LPHASE(L,LN,1)=60
  140 LOFFST(L,LN)=0
C
C     ******************************************************************
C
C          WRITE PROGRAM DATA
C
C     ******************************************************************
C
      WRITE (5,420) (MESAGE(I),I=1,8)
      WRITE (6,430) CYCLE,MINITIM,MLRUN
      DEC1=-DECMN(1)
      DEC2=-DECMN(2)
      WRITE (6,440) A0,FREEK,REACT,ACCMN(1),ACCSIG(1),ACCMN(2),ACCSIG(2)
     1,DEC1,DECSIG(1),DEC2,DECSIG(2),FLMN(1),FLSIG(1),FLMN(2),FLSIG(2),(
```

TABLE B-2, cont.

```
     2L,TVELMN(L,1),TVELSIG(L,1),TVELMN(L,2),TVELSIG(L,2),L=1,NOLINK)
      DO 150 L=1,NOLINK
      DO 150 I=1,2
      TVELMN(L,I)=TVELMN(L,I)*1.47
  150 TVELSIG(L,I)=TVELSIG(L,I)*1.47
      WRITE (6,450)
      DO 160 LINK=1,NOLINK
      NOL=NOLANE(LINK)
      WRITE (6,460) LINK
      DO 160 LANE=1,NOL
      WRITE (6,470) LANE,LCYCLE(LINK),NOPHAS(LINK,LANE),NPHASE(LINK,LANE
     1,1),LPHASE(LINK,LANE,1),LOFFST(LINK,LANE)
      NOP=NOPHAS(LINK,LANE)
      IF (NOP.EQ.1) GO TO 160
      WRITE (6,480) (NPHASE(LINK,LANE,K),LPHASE(LINK,LANE,K),K=2,NOP)
  160 CONTINUE
      NOTEC=1
      ISTART=NOLINK-NINPUT+1
      IEND=NOLINK
C
C     **************************************************************
C
C          READ AND WRITE TRAFFIC DATA
C
C     **************************************************************
C
      WRITE (6,490)
      DO 170 L=1,NOLINK
      WRITE (6,500) L,(NREC(L,J),LREC(L,J),J=1,[illegible]),(IOPR(L,J),J=1,5)
  170 CONTINUE
      WRITE (6,510)
      DO 180 L=1,NOLINK
  180 WRITE (6,520) L,STRLINE(L),(DRED-Y(L,I),I=1,[illegible]),(YLINK(L,I),I=1,3),
     1(RADIUS(L,I),I=1,6)
      DO 210 I=1,10
      IF (NEWPR(I).EQ.0) GO TO 190
      NEWPR(I)=NEWPR(I)*60+INITIM
  190 IF (NEWVOL(I).EQ.0) GO TO 200
      NEWVOL(I)=NEWVOL(I)*60+INITIM
  200 IF (NEWINIT(I).EQ.0) GO TO 210
      NEWINIT(I)=NEWINIT(I)*60+INITIM
  210 CONTINUE
      CALL INITRN
      WRITE (6,530)
      CALL INIVOL
C
C     **************************************************************
C
C          INITIALIZE
C
C     **************************************************************
C
      CALL INIT
      ITIME=0
      INIKNT=0
      DO 220 L=1,NQ
      DO 220 LN=1,3
  220 LQUEUE(L,LN)=0
      DO 230 L=1,NOLINK
      LINCAR(L)=0
      DO 230 K=1,3
```

TABLE B-2, cont.

```
      IFRSTCR(L,K)=0
      ILSTCAR(L,K)=0
  230 LANCAR(L,K)=0
      ITT=INITIM+LRUN+1
      INTTT=INT(CYCLE)
      SUMKNT=1
      KNTVOL=1
      KNTTRN=1
      KNTINIT=1
      LASTONE=0
      INEXT=1
      IREM=ISIZE
      IMAX=0
C
C     *********************************************************************
C
C          BEGIN TIME LOOP
C
C     *********************************************************************
C
      CUM=TIMELP(0)
      DO 310 KKTIM=1,ITT,INTTT
      ITIME=KKTIM-1
      IF (ITIME.LT.INITIM) GO TO 240
      IF (SUMKNT.GT.15) GO TO 240
      IF (SUMTIM(SUMKNT).EQ.0) GO TO 240
      IF (ITIME.LT.SUMTIM(SUMKNT)) GO TO 240
      CUM=TIMELP(1)
      CALL SUMARY (VOL,ITIME,SUMKNT)
      SUMKNT=SUMKNT+1
      CUM=TIMELP(1)
  240 CONTINUE
      IF (ITIME.LT.INITIM) GO TO 250
      IF (KNTTRN.GT.10) GO TO 250
      IF (NEWPR(KNTTRN).EQ.0) GO TO 250
      IF (ITIME.LT.NEWPR(KNTTRN)) GO TO 250
      KNTTRN=1+KNTTRN
      KTIM=ITIME/60-MINITIM
      WRITE (6,540) KTIM
      CALL INITRN
  250 IF (ITIME.LT.INITIM) GO TO 260
      IF (KNTVOL.GT.10) GO TO 260
      IF (NEWVOL(KNTVOL).EQ.0) GO TO 260
      IF (ITIME.LT.NEWVOL(KNTVOL)) GO TO 260
      KNTVOL=1+KNTVOL
      KTIM=ITIME/60-MINITIM
      WRITE (6,550) KTIM
      CALL INIVOL
  260 CONTINUE
      IF (ITIME.LT.INITIM) GO TO 270
      IF (KNTINIT.GT.10) GO TO 270
      IF (NEWINIT(KNTINIT).EQ.0) GO TO 270
      IF (ITIME.LT.NEWINIT(KNTINIT)) GO TO 270
      KNTINIT=KNTINIT+1
      CALL INIT
  270 CONTINUE
      IF (ITIME.GE.INITIM+1.AND.INIKNT.EQ.0) GO TO 290
      GO TO 230
C
C     *********************************************************************
C
```

TABLE B-2, cont.

```
C          RE-INITIALIZE FOR RUN-TIME
C
C     ******************************************************************
C
  280 CONTINUE
      INIKNT=1
      CALL INIT
  290 CONTINUE
C
C     ******************************************************************
C
C          SCAN AND UPDATE
C
C     ******************************************************************
C
      CALL INPLIN
      IF (IDEFST.EQ.0.AND.IDEFEND.EQ.0) GO TO 300
      IF (ITIME.LT.IDEFST) GO TO 300
      IF (ITIME.GT.IDEFEND) GO TO 300
      CALL DEFINE
  300 CONTINUE
      CALL UPDATE
  310 CONTINUE
C
C     ******************************************************************
C
C          SUMMARIZE RUN
C
C     ******************************************************************
C
      CUM=TIMELP(1)
      CALL SUMARY (VOL,ITIME,SUMKNT)
      WRITE (6,560) IMAX
      STOP
C
  320 FORMAT (8A10)
  330 FORMAT (F2.0,3I3,I4,I5,15I4)
  340 FORMAT (10I4)
  350 FORMAT (2I5,020)
  360 FORMAT (I2,15I3)
  370 FORMAT (I2,10I3)
  380 FORMAT (4F5.2)
  390 FORMAT (5X,7I1,17I3,4F3.0)
  400 FORMAT (5X,16F4.0,/5X,6F3.0)
  410 FORMAT (5X,I3,3I1,15I2,/5X,15I2,3I3)
  420 FORMAT (1H1,4X,125(1H*),/5X,125(1H*),/5X,125(1H*),////62X, 11HS I
     1G N E T,//51X, 33HA PROGRAM FOR THE SIMULATION OF A,/57X, 22HTRAFF
     2IC SIGNAL NETWORK,///28X,8A10,////5X,125(1H*),/5X,125(1H*),/5X,125
     3(1H*))
  430 FORMAT (/////10X, 13HSCAN CYCLE = ,F3.1,  8H SECONDS,///10X, 22HIN
     1ITIALIZATION TIME = ,I4,  8H MINUTES,20X, 11HRUN TIME = ,I5,  8H M
     2INUTES)
  440 FORMAT (////10X, 18HPROGRAM PARAMETERS,///10X, 23HCHARACTERISTIC S
     1PEED = ,F5.2, 16H FEET PER SECOND,//10X, 30HPROPORTIONALITY COEFFI
     2CIENT = ,F4.2,//10X, 16HREACTION TIME = ,F4.2,  8H SECONDS,//10X,
     328HMAXIMUM DESIRED ACCELERATION,/20X,  4HCARS,10X,  7HMEAN = ,F5.2
     4, 21H FEET PER SECOND SQD.,10X, 12HSTD. DEV. = ,F5.2, 21H FEET PER
     5 SECOND SQD.,/20X,  6HTRUCKS,8X,  7HMEAN = ,F5.2, 21H FEET PER SEC
     6OND SQD.,10X, 12HSTD. DEV. = ,F5.2, 21H FEET PER SECOND SQD.,//10X
     7, 28HMINIMUM DESIRED DECELERATION,/20X,  4HCARS,10X,  7HMEAN = ,F5
     8.2, 16H FEET PER SECOND,10X, 12HSTD. DEV. = ,F5.2, 21H FEET PER SE
```

TABLE B-2, cont.

```
      9COND SQD.,/20X,  6HTRUCKS,8X,  7HMEAN = ,F5.2, 16H FEET PER SECOND  A 3050
     *,10X, 12HSTD. DEV. = ,F5.2, 21H FEET PER SECOND SQD.,//10X, 16HEFF  A 3060
     *ECTIVE LENGTH,/20X,  4HCARS,10X,  7HMEAN = ,F5.2,  5H FEET,10X, 12  A 3070
     *HSTD. DEV. = ,F5.2,  5H FEET,/20X,  6HTRUCKS,9X,  7HMEAN = ,F5.2,   A 3080
     * 5H FEET,10X, 12HSTD. DEV. = ,F5.2,//10X, 15HTARGET VELOCITY,15X,   A 3090
     * 4HCARS,30X,  6HTRUCKS,/22X,  4HMEAN,9X,  9HSTD. DEV.,13X,  4HMEAN  A 3100
     *,9X,  9HSTD. DEV.,/(20X,  4HLINK,I3,4X,F6.2,  4H MPH,6X,F6.2,13X,F  A 3110
     *6.2,  4H MPH,6X,F6.2/))                                             A 3120
  450 FORMAT (1H2,9X, 18HSIGNAL TIMING DATA,///31X,  5HCYCLE,10X,  9HNUM  A 3130
     1BER OF,5X, 10HINDICATION,5X,  8HDURATION,10X,  6HOFFSET,/30X,  6HL  A 3140
     2ENGTH,12X,  6HPHASES,//)                                            A 3150
  460 FORMAT (10X,  4HLINK,I3)                                            A 3160
  470 FORMAT (19X,  4HLANE,I2,6X,I2,16X,I1,13X,I2,11X,I3,16X,I3)          A 3170
  480 FORMAT (4(64X,I2,11X,I3/))                                          A 3180
  490 FORMAT (/////10X, 19HRECEIVING LINK-LANE,61X, 13HOPPOSING LINK,//2  A 3190
     16X,  3HRT1,5X,  3HRT2,5X,  3HST1,5X,  3HST2,5X,  3HLT1,5X,  3HLT2,  A 3200
     223X,  1H1,6X,  1H2,6X,  1H3,6X,  1H4,6X,  1H5,///)                  A 3210
  500 FORMAT (10X,  4HLINK,I3,8X,6(I2,  1H-,I1,4X),20X,5(I2,5X))          A 3220
  510 FORMAT (/////10X, 13HLINK GEOMETRY,//23X,  4HCURB,6X, 12HBOUNDARY (1  A 3230
     1-6),29X,  9HLLAN(1-3),13X, 11HRADIUS(1-6),//)                       A 3240
  520 FORMAT (10X,  4HLINK,I3,6X,F4.0,5X,6(F4.0,2X),6X,3(F4.0,2X),5X,6(F  A 3250
     13.0,2X))                                                            A 3260
  530 FORMAT (/////10X, 22HINITIAL TARGET VOLUMES,/)                      A 3270
  540 FORMAT (/////10X, 44HNEW TURNING PROBABILITIES ENTERED AT TIME = ,  A 3280
     1I4,  8H MINUTES)                                                    A 3290
  550 FORMAT (/////10X, 37HNEW TARGET VOLUMES ENTERED AT TIME = ,I4,  8H  A 3300
     1 MINUTES)                                                           A 3310
  560 FORMAT (1H1,/////20X, 39HMAXIMUM NUMBER OF VEHICLES IN SYSTEM = ,I  A 3320
     16)                                                                  A 3330
C                                                                         A 3340
      END                                                                 A 3350


      SUBROUTINE INIT                                                     B   10
      COMMON /B2/ TTIME(84),TDIST(84),TRTIME(15,105),TTTIME,TTDIST,HSTO(  B   20
     115,3,14)                                                            B   30
      COMMON /B3/ DELAY(84),DELDEV(84),NCALL(84)                          B   40
      COMMON /QU/ LQUEUE(15,3),LQ(10),NQ                                  B   50
      COMMON /TRCE/ NTRACE,LKTRCE(15),NOTRC(15),INTTRC,KNTRAC(15)         B   60
      COMMON /KWNT7/ NCAR(84),NOUTPT,ISTART,IEND,KGEN(84)                 B   70
      COMMON /RELAT/ LINCAR(84),NOLINK,LANCAR(84,3),NOLANE(84)            B   80
      DIMENSION HTR(105)                                                  B   90
      TTTIME=0.                                                           B  100
      TTDIST=0.                                                           B  110
      DO 100 L=1,NOLINK                                                   B  120
      NCAR(L)=0                                                           B  130
      KGEN(L)=0                                                           B  140
      TTIME(L)=0.                                                         B  150
      TDIST(L)=0.                                                         B  160
  100 CALL STIX4 (DELAY(L),DELDEV(L),NCALL(L),VMEAN,STDEV)                B  170
      DO 120 L=1,NTRACE                                                   B  180
      CALL STIX1 (HTR,200.,0.,100,IT,11HTRAVEL TIME,11,LINK)              B  190
      KNTRAC(L)=0                                                         B  200
      NOTRC(L)=INTTRC                                                     B  210
      DO 110 I=1,105                                                      B  220
  110 TRTIME(L,I)=HTR(I)                                                  B  230
  120 CONTINUE                                                            B  240
      DO 130 L=1,NQ                                                       B  250
      LNK=LQ(L)                                                           B  260
      N=NOLANE(LNK)                                                       B  270
```

TABLE B-2, cont.

```
      DO 130 LN=1,N
      CALL STIX1 (HTR,10.,0.,10,DUM,12HQUEUE LENGTH,12,LINK)
      DO 130 I=1,14
  130 HSTO(L,LN,I)=HTR(I)
      RETURN

      END
```

```
      SUBROUTINE INITRN
      COMMON /PFLAT/ LINCAP(84),NOLINK,LANCAP(84,3),NOLANE(84)
      COMMON /TURN/ NREC(84,6),LREC(84,6),PTURN(84,6),FTRN(84,2)
      COMMON /KNNT2/ NCAR(84),NOUTPT,ISTART,IEND,KGEN(84)
      COMMON /LINKS/ LANTN(84,6),LLAN(84,3)
      REAL LLAN
      N=NOUTPT+1
      DO 100 L=N,NOLINK
  100 READ (5,210) (PTURN(L,J),J=1,6)
      WRITE (6,220)
      DO 130 I=1,NOUTPT
      NLNE=NOLANE(I)
      GO TO (110,120,130), NLNE
  110 PTURN(I,1)=0.
      PTURN(I,2)=0.
      PTURN(I,3)=1.
      PTURN(I,4)=1.
      PTURN(I,5)=1.
      PTURN(I,6)=1.
      GO TO 130
  120 PTURN(I,1)=0.
      PTURN(I,2)=0.
      PTURN(I,3)=0.5
      PTURN(I,4)=1.
      PTURN(I,5)=1.
      PTURN(I,6)=1.
  130 CONTINUE
      DO 140 L=N,NOLINK
      WRITE (6,230) L,(PTURN(L,J),J=1,6),(LANTN(L,J),J=1,6)
  140 CONTINUE
      NOUT=NOUTPT+1
      DO 200 LINK=NOUT,NOLINK
      IF (NOLANE(LINK).EQ.1) GO TO 200
      NOL=NOLANE(LINK)
      N=NREC(LINK,3)
      IF (N.EQ.0) GO TO 200
      DO 200 LANE=1,NOL
      IF (LANE.EQ.3) GO TO 200
      IF (PTURN(LINK,4).LE.PTURN(LINK,2)) GO TO 160
  150 PSTH=1.
      PRDIFF=0.
      GO TO 170
  160 CONTINUE
      PST=(PTURN(LINK,LANE+2)-PTURN(LINK,LANE+1))/(PTURN(LINK,4)-PTURN(L
     1INK,2))
      PSTH=PST
      IF (PSTH.EQ.0.) GO TO 150
      PST=PST-PTURN(N,2)*PST-(1.00-PTURN(N,4))*PST
      PRDIFF=PTURN(N,LANE+2)-PTURN(N,LANE+1)-PST
      IF (PRDIFF.GT.0.) PRDIFF=0.
  170 GO TO (180,190), LANE
```

TABLE B-2, cont.

```
180 PTRN(N,LANE)=PTURN(N,4)+FRDIFF/PSTH
    GO TO 200
190 PTRN(N,LANE)=PTURN(N,2)-FRDIFF/PSTH
200 CONTINUE
    RETURN

210 FORMAT (5X,6F4.2)
220 FORMAT (/////10X, 32HCUMULATIVE TURNING PROBABILITIES,37X, 19HLANE
   1 RECD. FOR TURN,//26X,  3HRT1,5X,  3HRT2,5X,  3HST1,5X,  3HST2,5X,
   2  3HLT1,5X,  3HLT2,15X,  3HRT1,5X,  3HRT2,5X,  3HST1,5X,  3HST2,5X
   3,  3HLT1,5X,  3HLT2,///)
230 FORMAT (10X,  4HLINK,I3,8X,F4.2,5(F8.2),15X,6(I3,5X))

    END



    SUBROUTINE INIVOL
    COMMON /LINKS/ LANEN(84,6),LLAN(84,3)
    COMMON /RELAT/ LINCAP(84),NOLINK,LANCAP(84,3),NOLANE(84)
    COMMON /GEOM/ CURB(84),BENDRY(84,6)
    COMMON /TIME/ FLTIME(84,3)
    COMMON /KWRIT2/ NCAR(84),NOUTPT,ISTART,IEND,KGEN(84)
    COMMON /TURN/ NREC(84,6),LREC(84,6),PTURN(84,6),PTRN(84,2)
    COMMON /VLME/ VOL(84),VOLUM(84,3)
    COMMON /TM/ ITIME
    COMMON /PARAM2/ TVELMN(84,2),TVELSIG(84,2),ACCMN(2),ACCSIG(2),DECM
   1N(2),DECSIG(2),FLMN(2),FLSIG(2),PRTRUCK(84)
    REAL LLAN
    DO 250 L=ISTART,IEND
    READ (5,260) VOL(L),PRTRUCK(L)
    WRITE (5,270) L,VOL(L),PRTRUCK(L)
    NOLLL=NOLANE(L)
    ILT=1
    P1=PTURN(L,3)
    DO 250 LAN=1,NOLLL
    IF (ITIME.EQ.0) FLTIME(L,LAN)=0.
    IF (FLTIME(L,LAN).EQ.10.**8.) FLTIME(L,LAN)=FLOAT(ITIME+3)
    IF (NOLLL.NE.1) GO TO 110
100 VOLUM(L,1)=VOL(L)/3600.
    GO TO 240
110 IF (LLAN(L,1).NE.CURB(L)) GO TO 150
    IF (LLAN(L,NOLLL).NE.CURB(L)) GO TO 140
    GO TO (120,130,230), LAN
120 VOLUM(L,1)=P1*VOL(L)/3600.
    GO TO 240
130 VOLUM(L,2)=(1.-P1)*VOL(L)/3600.
    GO TO 240
140 IF (LAN.EQ.NOLLL) GO TO 230
    IF (NOLANE(L).EQ.2) GO TO 100
    GO TO (120,130), LAN
150 GO TO (160,160,170), NOLLL
160 IF (LAN.EQ.1) GO TO 230
    VOLUM(L,LAN)=VOL(L)/3600.
    GO TO 240
170 GO TO (230,180,190), LAN
180 IF (CURB(L).EQ.LLAN(L,3)) GO TO 200
    GO TO (230,190,230), LAN
190 VOLUM(L,LAN)=VOL(L)/3600.
    GO TO 240
200 GO TO (230,210,220), LAN
```

TABLE B-2, cont.

```
 210 VOLUM(L,LAN)=P1*VOL(L)/3600.
     GO TO 240
 220 VOLUM(L,LAN)=(1.-P1)*VOL(L)/3600.
     GO TO 240
 230 VOLUM(L,LAN)=0.
 240 IF (VOLUM(L,LAN).EQ.0.) FLTIME(L,LAN)=10.**8.
 250 CONTINUE
     RETURN
C
 260 FORMAT (5X,F6.0,F4.2)
 270 FORMAT (10X,  4HLINK,I4,5X, 10H VOLUME = ,F6.0,10X, 17HPERCENT TRU
    1CKS = ,F4.2)
C
     END



     SUBROUTINE DEFINE
     COMMON /TM/ ITIME
     COMMON /RELAT/ LINCAR(84),NOLINK,LANCAR(84,3),NOLANE(84)
     COMMON /A1/ IREM,INEXT,IMAX,ISIZE,IFRSTCR(84,3),ILSTCAR(84,3),VEL(
    11500),POSN(1500),ACCEL(1500),ICAR(1500),NXTCAR(1500),IPRECAR(1500)
     COMMON /MSK/ MASK(20)
     IBITM(IX,M,N)=ISHFTR(IX,M).AND.MASK(N)
     WRITE (6,150) ITIME,IREM,INEXT
     DO 140 I=1,NOLINK
     WRITE (6,160) I,LINCAR(I)
     IF (LINCAR(I).EQ.0) GO TO 120
     NOLLL=NOLANE(I)
     DO 130 J=1,NOLLL
     WRITE (6,170) J,LANCAR(I,J),IFRSTCR(I,J),ILSTCAR(I,J)
     IF (LANCAR(I,J).EQ.0) GO TO 100
     NOC=LANCAR(I,J)
     IV=IFRSTCR(I,J)
     DO 110 K=1,NOC
     IT1=IBITM(ICAR(IV),33,3)
     IT2=IBITM(ICAR(IV),36,4)
     WRITE (6,180) K,POSN(IV),VEL(IV),ACCEL(IV),IT1,IT2
     WRITE (6,190) IV,NXTCAR(IV),IPRECAR(IV)
     IV=NXTCAR(IV)
 100 CONTINUE
 110 CONTINUE
 120 CONTINUE
 130 CONTINUE
 140 CONTINUE
     RETURN
C
 150 FORMAT (/////3X,5(1H*),  8H TIME = ,I5,2X,15(1H*),20X,  5HIREM=,I4
    1,10X,  6HINEXT=,I4)
 160 FORMAT (///5X,  4HLINK,I3,  6H CARS=,I3)
 170 FORMAT (//15X,  4HLANE,I3,  6H CARS=,I3,10X, 10HFIRST CAR=,I3,10X,
    1  9HLAST CAR=,I3/)
 180 FORMAT (25X,  4HVEH.,I3,4X,  5HPOSN=,F6.1,10X,  4HVEL=,F6.2,10X,
    14HACC=,F6.2,10X,  6HITURN=,I3,10X,  7HACCTYP=,I3)
 190 FORMAT (36X,  6HINDEX=,I3,10X,  9HNEXT CAR=,I3,10X, 10HPREV. CAR=,
    1I3)
C
     END
```

TABLE B 2, cont.

```
      SUBROUTINE INPLIN
      COMMON /A1/ IREM,INEXT,IMAX,ISIZE,IFRSTCR(84,3),ILSTCAR(84,3),VEL(
     11500),POSN(1500),ACCEL(1500),ICAR(1500),NXTCAR(1500),IPRECAR(1500)
      COMMON /A2/ IHOLE(1500),LASTONE
      COMMON /RELAT/ LINCAR(84),NOLINK,LANCAR(84,3),NOLANE(84)
      COMMON /TURN/ NREC(84,6),LREC(84,6),PTURN(84,6),PTP(84,2)
      COMMON /TIME/ FLTIME(84,3)
      COMMON /TRCE/ ITRACE,LKTRCE(15),NOTRC(15),INITRC,KNTRAC(15)
      COMMON /KWRIT2/ NCAR(84),NOUTPT,ISTART,IEND,KGEN(84)
      COMMON /TM/ ITIME
      COMMON /VLME/ VOL(84),VOLUM(84,3)
      COMMON /LINKS/ LANIN(84,6),LLAN(84,3)
      COMMON /GEOM/ CURB(84),DENDRY(84,6)
      COMMON /CAPACTY/ LNCAP
      COMMON /PARAM/ DUMMY(3),CYCLE
      COMMON /PARAM2/ TVELMN(84,2),TVELSIG(84,2),ACCMN(2),ACCSIG(2),DECM
     1N(2),DECSIG(2),FLMN(2),FLSIG(2),PRTRUCK(84)
      COMMON /MSK/ MASK(20)
      REAL LLAN
      REAL NORMAL,NEGEXP
      IBITR(IX,M,N)=ISHFTR(IX,M).AND.MASK(N)
      DO 340 I=ISTART,IEND
      NLANE=NOLANE(I)
      ILT=1
      IF (LLAN(I,NLANE).NE.CURB(I)) ILT=2
      IRT=1
      IF (LLAN(I,1).NE.CURB(I)) IRT=2
      DO 330 K=1,NLANE
      FLT=FLOAT(ITIME)
      IF (FLTIME(I,K).GE.FLT+CYCLE) GO TO 330
  100 KK=LANCAR(I,K)+1
      IF (IREM.LE.0.AND.LASTONE.EQ.0) GO TO 350
      IF (KK.EQ.1) GO TO 110
      IL=ILSTCAR(I,K)
      IF (POSN(IL).LT.-110.) GO TO 330
  110 CONTINUE
      IF (LASTONE.EQ.0) GO TO 120
      INDEX=IHOLE(LASTONE)
      LASTONE=LASTONE-1
      GO TO 130
  120 INDEX=INEXT
      INEXT=INEXT+1
      IREM=IREM-1
  130 CONTINUE
      IPRECAR(INDEX)=ILSTCAR(I,K)
      IF (LANCAR(I,K).EQ.0) IPRECAR(INDEX)=0
      NXTCAR(INDEX)=0
      LINDEX=IPRECAR(INDEX)
      IF (IPRECAR(INDEX).NE.0) NXTCAR(LINDEX)=INDEX
      ILSTCAR(I,K)=INDEX
      IS=ISIZE-IREM-LASTONE
      IMAX=MAX0(IS,IMAX)
      IF (LANCAR(I,K).EQ.0) IFRSTCR(I,K)=INDEX
      LANCAR(I,K)=LANCAR(I,K)+1
      LINCAR(I)=LINCAR(I)+1
      KGEN(I)=KGEN(I)+1
      RN=RANF(0)
      IVLTYP=1
      IF (RN.LE.PRTRUCK(I)) IVLTYP=2
      ITARVE=NORMAL(TVELMN(I,IVLTYP),TVELSIG(I,IVLTYP))
      IFL=NORMAL(FLMN(IVLTYP),FLSIG(IVLTYP))
```

TABLE B-2, cont.

```
      MAXDAC=NORMAL(ACCMN(IVLTYP),ACCSIG(IVLTYP))
      IDDEAC=NORMAL(DECMN(IVLTYP),DECSIG(IVLTYP))
      IACCTY=7
      RN=RANF(0)
      IF (IRT.EQ.1) GO TO 150
      IF (ILT.EQ.1) GO TO 140
      GO TO 180
  140 GO TO (200,200,220), K
  150 CONTINUE
      IF (ILT.EQ.1) GO TO 160
      GO TO 170
  160 IF (NLANE.LT.2) GO TO 180
      IF (K.EQ.1) GO TO 200
      GO TO 220
  170 IF (NLANE.LT.3) GO TO 180
      IF (K.EQ.1) GO TO 200
      GO TO 220
  180 DO 190 L=1,6
      JTURN=L
      IF (RN.LE.PTURN(I,L)) GO TO 240
  190 CONTINUE
      GO TO 240
  200 RN=RN*PTURN(I,3)
      DO 210 L=1,3
      JTURN=L
      IF (RN.LE.PTURN(I,L)) GO TO 240
  210 CONTINUE
      GO TO 240
  220 RN=RN*(1.-PTURN(I,3))+PTURN(I,3)
      DO 230 L=4,6
      JTURN=L
      IF (RN.LE.PTURN(I,L)) GO TO 240
  230 CONTINUE
  240 IF (I.EQ.LKTRCE(1)) GO TO 250
      CALL JBITM (ICAR(INDEX),56,1,0)
      GO TO 280
  250 IF (LKTRCE(2).EQ.NREC(I,JTURN)) GO TO 260
      CALL JBITM (ICAR(INDEX),56,1,0)
      GO TO 280
  260 IF (INTTRC.EQ.NOTRC(1)) GO TO 270
      NOTRC(1)=NOTRC(1)+1
      CALL JBITM (ICAR(INDEX),56,1,0)
      GO TO 280
  270 NOTRC(1)=0
      CALL JBITM (ICAR(INDEX),56,1,1)
  280 CONTINUE
      CALL JBITM (ICAR(INDEX),26,7,ITARVE)
      CALL JBITM (ICAR(INDEX),50,4,MAXDAC)
      CALL JBITM (ICAR(INDEX),46,4,IDDEAC)
      CALL JBITM (ICAR(INDEX),40,6,IEL)
      CALL JBITM (ICAR(INDEX),0,13,ITIME)
      CALL JBITM (ICAR(INDEX),54,2,IVLTYP)
      CALL JBITM (ICAR(INDEX),36,4,IACCTY)
      CALL JBITM (ICAR(INDEX),33,3,JTURN)
      IF (KK.EQ.1) GO TO 290
      JEL=IBITM(ICAR(LINDEX),40,6)
      IF (POSN(LINDEX)-FLOAT(JEL)-20..GT.0.) GO TO 290
      POSN(INDEX)=POSN(LINDEX)-FLOAT(JEL)-20.
      GO TO 300
  290 POSN(INDEX)=-(FLTIME(I,K)-FLOAT(ITIME))*FLOAT(ITARVE)
  300 CONTINUE
```

TABLE B-2, cont.

```
      IST=INT(ABS(POSN(INDEX)))
      CALL JRITM (ICAR(INDEX),14,9,IST)
      ACCEL(INDEX)=0.
      IF (KK.EQ.1) GO TO 310
      DELS=POSN(LINDEX)-POSN(INDEX)-FLOAT(JEL)
      IF (DELS.GT.100.) GO TO 310
      IF (VEL(LINDEX).GT.FLOAT(ITARVE)) GO TO 310
      VEL(INDEX)=FLOAT(ITARVE)-(100.-DELS)/100.*(FLOAT(ITARVE)-VEL(LINDE
     1X))
      GO TO 320
  310 VEL(INDEX)=FLOAT(ITARVE)
  320 CONTINUE
      IF (VOLUM(I,K).EQ.1.) GO TO 330
      FLTIME(I,K)=FLTIME(I,K)+NEGEXP(VOLUM(I,K))
      IF (FLTIME(I,K).LE.FLT) GO TO 100
  330 CONTINUE
  340 CONTINUE
      RETURN
  350 WRITE (6,360)
      RETURN
C
  360 FORMAT (/10X, 14HTABLE OVERFLOW)
C
      END
```

```
      SUBROUTINE UPDATE
      COMMON /A1/ IREM,INEXT,IMAX,ISIZE,IFRSTCAR(84,3),ILSTCAR(84,3),VEL(
     11500),POSN(1500),ACCEL(1500),ICAR(1500),NXTCAR(1500),IPRECAR(1500)
      COMMON /TM/ ITIME
      COMMON /PARAM/ A0,FREEK,REACT,CYCLE
      COMMON /LINKS/ LANTN(84,6),LLAN(84,3)
      COMMON /BUK/ NHOLD(84,3)
      COMMON /KHNTZ/ NCAR(84),NOUTPT,ISTART,IEND,KGEN(84)
      COMMON /PV/ P(84,3),V(84,3)
      COMMON /GEOM/ CURB(84),BRNDRY(84,6)
      COMMON /RELAT/ LINCAR(84),NOLINK,LANCAR(84,3),NOLANE(84)
      COMMON /CAR/ IACC,TVEL,ITURN,FL,IACCL
      COMMON /STARTS/ T2,MGO
      COMMON /MSK/ MASK(20)
      INTEGER ACCTYP
      REAL LLAN
C
C     1=STOPPED                                  7=FREE BEHAVIOR
C     2=STOPPING                                 8=CAR FOLLOWING
C     3=STOPPED FOR TURN                         9=IN DELAYED LEFT TURN
C     4=STOPPING FOR TURN                       10=IN NORMAL TURN
C     5=STOPPING FOR LANE CHANGE                11=SLOWING FOR TURN
C     6=STOPPED FOR LANE CHANGE                 12=STOPPING FOR RED SIGNAL
C
      IBITM(IX,M,N)=ISHFTR(IX,M).AND.MASK(N)
      DO 100 I=1,NOLINK
      NLAN=NOLANE(I)
      DO 100 J=1,NLAN
      NHOLD(I,J)=0
  100 CONTINUE
      THOK=REACT
C
C     ******************************************************************
C
```

TABLE B-2, cont.

```
C           CHANGE LANES
C
C
      CALL CHANGE
C
C     ******************************************************************
C
C           UPDATE VEHICLES
C
C
      DO 430 LINK=1,NOLINK
      IF (LINCAR(LINK).EQ.0) GO TO 430
      NOLL=NOLANE(LINK)
      DO 430 LANE=1,NOLL
      IF (LANCAR(LINK,LANE).EQ.0) GO TO 430
      CALL SIGNAL (ISIG,LINK,LANE,ITIME)
      LANNN=LANCAR(LINK,LANE)
      NUM=IFRSTCR(LINK,LANE)
  110 MGO=0
      IACCL=IACC
      IACC=IBITM(ICAR(NUM),36,4)
      TVFL=FLOAT(IBITM(ICAR(NUM),26,7))
      ITURN=IBITM(ICAR(NUM),33,3)
      IF (LINK.GT.NOUTPT) GO TO 120
      IF (NUM.NE.IFRSTCR(LINK,LANE)) GO TO 120
      ACCFL(NUM)=FREEK*(TVFL-VFL(NUM))
      ACCTYP=7
      GO TO 330
  120 CONTINUE
      ACCPR=ACCFL(NUM)
      ICHAGH=0
      IF (IACC.EQ.13) GO TO 310
      IF (IACC.NE.5.AND.IACC.NE.6.OR.LANE.EQ.LANTN(LINK,ITURN)) GO TO 13
     10
      ICHAGH=1
      IS=ISIG
      IF (IS.GT.2) IS=2
      GO TO (140,310), IS
  130 GO TO (140,160,170,180,190,200,210,220,230,240,250,260,270,150), I
     1SIG
  140 CALL REDAPP (LINK,LANE,NUM,ACCTYP)
      GO TO 330
  150 CALL AMBER (IGO,LINK,LANE,NUM,ACCTYP)
      IF (IGO.EQ.0) GO TO 140
  160 GO TO (280,280,300,300,290,290), ITURN
  170 IF (ITURN.NE.3.AND.ITURN.NE.4) GO TO 140
      GO TO 300
  180 IF (ITURN.NE.6) GO TO 140
      GO TO 290
  190 IF (ITURN.NE.1) GO TO 140
      GO TO 280
  200 IF (ITURN.NE.5) GO TO 140
      GO TO 290
  210 IF (ITURN.NE.2) GO TO 140
      GO TO 280
  220 IF (ITURN.NE.3.AND.ITURN.NE.4.AND.ITURN.NE.6) GO TO 140
      IF (ITURN.EQ.6) GO TO 290
      GO TO 300
  230 IF (ITURN.NE.3.AND.ITURN.NE.4.AND.ITURN.NE.1) GO TO 140
      IF (ITURN.EQ.1) GO TO 280
      GO TO 300
```

TABLE B-2, cont.

```
240 IF (ITURN.NE.3.AND.ITURN.NE.4.AND.ITURN.NE.5) GO TO 140
    IF (ITURN.EQ.5) GO TO 290
    GO TO 300
250 IF (ITURN.NE.3.AND.ITURN.NE.4.AND.ITURN.NE.2) GO TO 140
    IF (ITURN.EQ.2) GO TO 280
    GO TO 300
260 IF (ITURN.LE.4) GO TO 140
    GO TO 230
270 IF (ITURN.GE.3) GO TO 140
280 CONTINUE
    CALL RIGHT (LINK,LANE,NUM,ACCTYP)
    GO TO 330
290 CONTINUE
    CALL LEFT (LINK,LANE,NUM,ITIME,ACCTYP)
    GO TO 330
300 CONTINUE
    IF (NUM.NE.IFRSTCR(LINK,LANE)) GO TO 310
    IF (POSN(NUM).LT.CURB(LINK)-100.) GO TO 310
    CALL CONST (LINK,LANE,NUM,ICON,0)
    IF (ICON.EQ.1) GO TO 320
310 CALL GENBHR (LINK,LANE,NUM,ACCTYP)
    GO TO 330
320 IF (VEL(NUM).LT.0.) VEL(NUM)=0.
    IF (POSN(NUM).GE.CURB(LINK)) GO TO 310
    XIX=CURB(LINK)-POSN(NUM)
    IF (XIX.EQ.0.) XIX=.5
    ACCEL(NUM)=-(VEL(NUM)*VEL(NUM))/XIX
    ACCTYP=2
330 CONTINUE
    IF (ICHNGH.NE.1) GO TO 340
    IF (ACCEL(NUM).LT.ACCPR) GO TO 340
    ACCEL(NUM)=ACCPR
    ACCTYP=IACC
340 CONTINUE
    IF (NUM.EQ.IFRSTCR(LINK,LANE).AND.IACC.EQ.1.AND.ACCTYP.NE.1) REACT
   1=.35
    P(LINK,LANE)=POSN(NUM)
    V(LINK,LANE)=VEL(NUM)
    IF (MGO.EQ.0) GO TO 350
    VEL(NUM)=ACCEL(NUM)*(CYCLE-T2)
    IF (VEL(NUM).LT.0.) VEL(NUM)=0.
    TT3=CYCLE-T2
    POSN(NUM)=POSN(NUM)+.5*ACCEL(NUM)*TT3*TT3
    MGO=0
    GO TO 400
350 V2=VEL(NUM)+ACCPR*REACT
    VEL(NUM)=V2+ACCEL(NUM)*(CYCLE-REACT)
    IF (VEL(NUM).GE.0.) GO TO 380
    IF (V2.GE.0.) GO TO 360
    T=-V(LINK,LANE)/ACCPR
    ABST=ABS(T)
    POSN(NUM)=POSN(NUM)+V(LINK,LANE)*T+.5*ACCPR*ABST*ABST
    GO TO 370
360 T=-V2/ACCEL(NUM)
    ABST=ABS(T)
    POSN(NUM)=POSN(NUM)+V2*T+.5*ACCPR*ABST*ABST
370 ACCEL(NUM)=0.
    VEL(NUM)=0.
    ACCTYP=1
    IF (ICHNGH.EQ.1) ACCTYP=6
    GO TO 400
```

TABLE B-2, cont.

```
380 POSN(NUM)=P(LINK,LANE)+V(LINK,LANE)*REACT+.5*ACCDR*REACT*REACT+V2*
   1(CYCLE-REACT)+.5*ACCEL(NUM)*(CYCLE-REACT)*(CYCLE-REACT)
    IF (POSN(NUM).LT.P(LINK,LANE)) POSN(NUM)=P(LINK,LANE)
    IF (NUM.EQ.IFRSTCR(LINK,LANE)) GO TO 390
    N=IPRECAR(NUM)
    IF (POSN(NUM).GT.POSN(N)) POSN(NUM)=POSN(N)-20.
390 CONTINUE
    IF (VEL(NUM).LT.0.) VEL(NUM)=0.
400 CONTINUE
    IF (ACCTYP.EQ.0) GO TO 410
    CALL JRITM (ICAR(NUM),36,4,ACCTYP)
410 CONTINUE
    IF (ACCTYP.EQ.0) ACCTYP=IACC
    REACT=THOK
    CALL BOOK (LINK,LANE,NUM,ITIME)
    IF (NUM.EQ.ILSTCAR(LINK,LANE)) GO TO 420
    NUM=NXTCAR(NUM)
    IF (NUM.NE.0) GO TO 110
420 CONTINUE
    CALL QUEUE (LINK,LANE)
430 CONTINUE
    CALL OUTPT (ITIME)
    RETURN
C
    END


    SUBROUTINE SIGNAL (ISIG,LINK,LANE,ITIME)
    COMMON /SIG/ LCYCLE(84),NOPHAS(84,3),NPHASE(84,3,5),LPHASE(84,3,5)
   1,LOFFST(84,3)
    INTEGER OFFPT
C
C   1= RED            6= GADL          11= GAS+GADR
C   2= GREEN          7= GADR          12= GAL+GADL
C   3= GAS            8= GAS+GAL       13= GAR+GADR
C   4= GAL            9= GAS+GAR       14= AMBER
C   5= GAR           10= GAS+GADL
C
    IF (ITIME.LT.LOFFST(LINK,LANE)) GO TO 100
    NCYC=(ITIME-LOFFST(LINK,LANE))/LCYCLE(LINK)
    OFFPT=NCYC*LCYCLE(LINK)+LOFFST(LINK,LANE)
    GO TO 110
100 NCYC=0
    OFFPT=-LCYCLE(LINK)+LOFFST(LINK,LANE)
110 CONTINUE
    NOPP=NOPHAS(LINK,LANE)
    DO 120 I=1,NOPP
    IT=OFFPT+LPHASE(LINK,LANE,I)
    IHOLD=I
    IF (ITIME.LE.IT) GO TO 130
    OFFPT=IT
120 CONTINUE
130 CONTINUE
    ISIG=NPHASE(LINK,LANE,IHOLD)
    RETURN
C
    END
```

TABLE B-2, cont.

```
      SUBROUTINE AMBER (IGO,LINK,LANE,NUM,ACCTYP)
      COMMON /A1/ IREM,INEXT,IMAX,ISIZE,IFRSTCR(84,3),ILSTCAR(84,3),VEL(
     11500),POSN(1500),ACCEL(1500),ICAR(1500),NXTCAR(1500),IPRECAR(1500)
      COMMON /GEOM/ CURB(84),DPNDRY(84,6)
      COMMON /MSK/ MASK(20)
      COMMON /CAR/ IACC,TVEL,ITURN,FL,IACCL
      DIMENSION PRSTOP(10)
      REAL NORMAL
      INTEGER ACCTYP
      DATA (PRSTOP(I),I=1,10)/.008,.023,.054,.113,.207,.340,.49,.65,.78,
     1.98/
      IF (IACC.NE.2.AND.IACC.NE.12) GO TO 100
      IF (ACCEL(NUM).GE.0.) GO TO 100
      XFINAL=-VEL(NUM)*VEL(NUM)/(2.*ACCEL(NUM))+POSN(NUM)
      IF (XFINAL.LE.CURB(LINK)) GO TO 140
  100 CONTINUE
      IF (IACC.NE.1) GO TO 110
      IF (POSN(NUM).GT.CURB(LINK)) GO TO 130
      GO TO 140
  110 CONTINUE
      IF (NUM.EQ.IFRSTCR(LINK,LANE)) GO TO 120
      N=IPRECAR(NUM)
      IF (ACCTYP.LE.6.OR.ACCTYP.EQ.12) GO TO 140
  120 CONTINUE
      XFINAL=CURB(LINK)
      V2=VEL(NUM)+ACCEL(NUM)*REACT
      IF (V2.LT.0.) V2=0.
      P2=POSN(NUM)+VEL(NUM)*REACT+.5*ACCEL(NUM)*REACT*REACT
      XX=XFINAL-P2
      IF (XX.LE.0.) GO TO 130
      ACC=-(V2*V2)/(2.*XX)
      RN=RANF(0)
      KACC=ABS(ACC)/2.
      IF (KACC.LT.1) GO TO 140
      IF (KACC.GT.10) GO TO 130
      IF (RN.GT.PRSTOP(KACC)) GO TO 140
  130 CONTINUE
      IGO=1
C
C        IGO = 1 MEANS GO
C
      RETURN
  140 CONTINUE
      IGO=0
C
C        IGO = 0 MEANS STOP
      RETURN
C
      END


      SUBROUTINE CHANGE
      COMMON /A1/ IREM,INEXT,IMAX,ISIZE,IFRSTCR(84,3),ILSTCAR(84,3),VEL(
     11500),POSN(1500),ACCEL(1500),ICAR(1500),NXTCAR(1500),IPRECAR(1500)
      COMMON /GEOM/ CURB(84),DPNDRY(84,6)
      COMMON /LINKS/ LANTN(84,6),LLAN(84,3)
      COMMON /RELAT/ LINCAR(84),NOLINK,LANCAR(84,3),NOLANE(84)
      COMMON /PARAM/ A0,FREEK,REACT,CYCLE
      COMMON /CAPACTY/ LNCAP
      COMMON /TM/ ITIME
```

TABLE B-2, cont.

```
      COMMON /MSK/ MASK(20)
      DIMENSION TLAG(21)
      REAL LLAN
      DATA (TLAG(I),I=1,21)/0.,0.,0.,0.,.09,.18,.31,.49,.66,.81,.90,.96,
     1.985,.995,.999,.999,1.0,1.0,1.0,1.0,1.0/
      IBITM(IX,M,N)=ISHFTR(IX,M).AND.MASK(N)
      DO 600 LINK=1,NOLINK
      NLN=NOLANE(LINK)
      DO 600 LANE=1,NLN
      IF (LANCAR(LINK,LANE).EQ.0) GO TO 600
      INDEX=IFRSTCR(LINK,LANE)
  100 CONTINUE
      IF (LANCAR(LINK,LANE).LE.0) GO TO 610
      ITURN=IBITM(ICAR(INDEX),33,3)
      LAN=LANTN(LINK,ITURN)
      IF (LAN.GT.3) LAN=3
      IF (LAN.LT.1) LAN=1
      IF (LANE-LAN) 120,110,130
  110 CONTINUE
      GO TO 590
  120 LANNEW=LANE+1
      GO TO 140
  130 LANNEW=LANE-1
  140 CONTINUE
      END=CURB(LINK)-LLAN(LINK,LANNEW)
      EL=FLOAT(IBITM(ICAR(INDEX),40,6))
      ENDLAN=END+EL
      IF (POSN(INDEX).LT.ENDLAN) GO TO 550
      IF (LANCAR(LINK,LANNEW).NE.0) GO TO 150
      N=0
      GO TO 440
  150 CONTINUE
      X=POSN(INDEX)-EL
      INDADJ=IFRSTCR(LINK,LANNEW)
  160 IF (POSN(INDADJ).GT.X) GO TO 170
      N=INDADJ
C           N=INDEX OF FIRST CAR BEHIND FRONT OF CHANGE ZONE
      GO TO 440
  170 CONTINUE
      IF (NXTCAR(INDADJ).EQ.0) GO TO 180
      INDADJ=NXTCAR(INDADJ)
      GO TO 160
  180 CONTINUE
      N=0
      GO TO 440
      X=POSN(INDEX)-25.
      IF (POSN(INDADJ).GE.X) GO TO 190
      DV=VEL(INDADJ)-VEL(INDEX)
      IF (DV.LE.0.) GO TO 410
      DT=(X-POSN(INDADJ))/DV
      IF (DT.GT.10.) GO TO 410
      LAG=DT*2.+1.
      TL=RANF(0)
      IF (TL.LE.TLAG(LAG)) GO TO 410
  190 CONTINUE
      X=CURB(LINK)-300.
      IF (POSN(INDEX).GT.X) GO TO 200
      GO TO 590
  200 CONTINUE
      KACC=IBITM(ICAR(INDEX),36,4)
      IACC=KACC
```

TABLE B-2, cont.

```
      IF (KACC.NE.6) GO TO 280
      ACCEL(INDEX)=0.
      GO TO (210,220,210), LANE
  210 LADJ=2
      GO TO 230
  220 LADJ=1
      GO TO 230
  230 IF (LANE.LT.LADJ) GO TO 270
      INDADJ=IFRSTCR(LINK,LADJ)
  240 IF (INDADJ.EQ.0) GO TO 270
      IF (POSN(INDEX).LT.POSN(INDADJ)-FLOAT(IBITM(ICAR(INDADJ),40,6))-10
     1.) GO TO 250
      GO TO 260
  250 CONTINUE
      INDADJ=NXTCAR(INDADJ)
      GO TO 240
  260 IF (IBITM(ICAR(INDADJ),36,4).NE.6) GO TO 270
      IF (KACC.NE.13) CALL JBITM (ICAR(INDEX),36,4,13)
C            IACC = 13 IS TO BYPASS THE NORMAL EFFECT OF A
C                     RED SIGNAL IN ADJUSTMENT FOR LANE CHANGE
      GO TO 590
  270 IF (KACC.NE.6) CALL JBITM (ICAR(INDEX),36,4,6)
      GO TO 590
  280 CONTINUE
      IF (KACC.EQ.5) GO TO 290
      ACCEL(INDEX)=-FLOAT(IBITM(ICAR(INDEX),46,4))*2.0
  290 PF=POSN(INDEX)-VEL(INDEX)*VEL(INDEX)/(2.*ACCEL(INDEX))
      GO TO (300,310,320), LANE
  300 LADJ=2
      GO TO 330
  310 LADJ=1
      GO TO 330
  320 LADJ=2
  330 INDADJ=IFRSTCR(LINK,LADJ)
      IF (INDADJ.EQ.0) GO TO 400
  340 CONTINUE
      IF (PF.GT.POSN(INDADJ)+EL+10.) GO TO 350
      IF (PF.LT.POSN(INDADJ)-40.) GO TO 360
      ELADJ=FLOAT(IBITM(ICAR(INDADJ),40,6))
      IF (PF.LT.POSN(INDADJ)-ELADJ-10.) GO TO 360
      IF (IBITM(ICAR(INDADJ),36,4).NE.6) GO TO 400
      X=POSN(INDADJ)-ELADJ-10.-POSN(INDEX)
      IF (X.LE.0.) GO TO 370
      ACCEL(INDEX)=-VEL(INDEX)*VEL(INDEX)/(2.*X)
      PF=POSN(INDADJ)-ELADJ-10.
  350 IF (LANE.LT.LADJ) GO TO 400
      IF (ACCEL(INDADJ).GE.0.) GO TO 360
      PFADJ=POSN(INDADJ)-VEL(INDADJ)*VEL(INDADJ)/(2.*ACCEL(INDADJ))
      IF (PF-EL-10..GT.PFADJ) GO TO 400
      PFLZN=PFADJ-10.-FLOAT(IBITM(ICAR(INDADJ),40,6))
      IF (PF.LT.PFLZN) GO TO 360
      IF (POSN(INDEX).GE.PFLZN) GO TO 400
      ACCEL(INDEX)=-VEL(INDEX)*VEL(INDEX)/(2.*(PFLZN-POSN(INDEX)))
  360 CONTINUE
      INDADJ=NXTCAR(INDADJ)
      IF (INDADJ.NE.0) GO TO 340
  370 IF (INDADJ.EQ.IFRSTCR(LINK,LADJ)) GO TO 390
      LINDADJ=IPRECAR(INDADJ)
      IF (POSN(LINDADJ)-POSN(INDADJ).LT.EL+10.+25.) GO TO 400
  380 IF (KACC.NE.7) CALL JBITM (ICAR(INDEX),36,4,7)
      GO TO 590
```

TABLE B-2, cont.

```
  390 IF (CURB(LINK)-POSN(INDADJ).GE.FL+10.) GO TO 380
  400 CONTINUE
      KACC=5
      IF (IACC.NE.KACC) CALL JBITM (ICAR(INDEX),36,4,KACC)
      GO TO 590
C           CHANGE LANES
  410 IF (N.EQ.IFRSTCR(LINK,LANNEW)) GO TO 440
      IF (LANCAR(LINK,LANNEW).EQ.0) GO TO 440
      NL=IPRECAR(INDADJ)
      IF (N.EQ.0) NL=ILSTCAR(LINK,LANNEW)
      IF (ACCEL(NL).LT.0.) GO TO 420
      T=CYCLE+REACT
      P2L=POSN(NL)+VEL(NL)*T+.5*ACCEL(NL)*T*T
      P2N=POSN(INDEX)+VEL(INDEX)*REACT+.5*ACCEL(INDEX)*REACT*REACT+(VEL(
     1INDEX)+ACCEL(INDEX)*REACT)*CYCLE+.5*(-7.)*CYCLE*CYCLE
      GO TO 430
  420 P2L=POSN(NL)-VEL(NL)*VEL(NL)/(2.*ACCEL(NL))
      P2N=POSN(INDEX)*VEL(INDEX)*VEL(INDEX)
  430 CONTINUE
      IF (P2N.GT.P2L-FLOAT(IBITM(ICAR(NL),40,6))) GO TO 190
  440 IOL=IPRECAR(INDEX)
      IOF=NXTCAR(INDEX)
      LANCAR(LINK,LANNEW)=LANCAR(LINK,LANNEW)+1
      LANCAR(LINK,LANE)=LANCAR(LINK,LANE)-1
      LA1=LANCAR(LINK,LANNEW)
      LA=LA1
      IF (N.NE.0) GO TO 460
      IL=ILSTCAR(LINK,LANNEW)
      IF (IL.EQ.0) GO TO 450
      NXTCAR(IL)=INDEX
  450 IPRECAR(INDEX)=IL
      NXTCAR(INDEX)=0
      ILSTCAR(LINK,LANNEW)=INDEX
      IF (LANCAR(LINK,LANNEW).EQ.1) IFRSTCR(LINK,LANNEW)=INDEX
      N=LA1
      GO TO 480
  460 CONTINUE
      IL=IPRECAR(N)
      IF=N
      IF (IL.EQ.0) GO TO 470
      NXTCAR(IL)=INDEX
  470 NXTCAR(INDEX)=IF
      IPRECAR(INDEX)=IL
      IPRECAR(IF)=INDEX
      IF (IPRECAR(INDEX).EQ.0) IFRSTCR(LINK,LANNEW)=INDEX
  480 CONTINUE
      IF (IFRSTCR(LINK,LANE).EQ.INDEX) IFRSTCR(LINK,LANE)=IOF
      IF (IFRSTCR(LINK,LANE).EQ.IOF) IPRECAR(IOF)=0
      IF (ILSTCAR(LINK,LANE).EQ.INDEX) ILSTCAR(LINK,LANE)=IOL
      IF (ILSTCAR(LINK,LANE).EQ.IOL) NXTCAR(IOL)=0
      IF (IBITM(ICAR(INDEX),36,4).EQ.6) GO TO 490
      CALL JBITM (ICAR(INDEX),36,4,8)
      GO TO 500
  490 CALL JBITM (ICAR(INDEX),36,4,13)
  500 CONTINUE
      IF (IOL.EQ.0) GO TO 510
      NXTCAR(IOL)=IOF
  510 IF (IOF.EQ.0) GO TO 520
      IPRECAR(IOF)=IOL
  520 CONTINUE
  530 IF (IL.EQ.0) GO TO 540
```

TABLE B-2, cont.

```
      IF=NXTCAR(IL)
      IF (IF.EQ.0) GO TO 540
      PX=POSN(IL)-FLOAT(IBITM(ICAR(IL),40,6))-5.
      IF (PX-POSN(IF).LE.0.) POSN(IF)=PX
      IL=NXTCAR(IL)
      GO TO 530
  540 INDEX=IOF
      IF (IOF.NE.0) GO TO 100
      GO TO 600
  550 CONTINUE
      DT=ENDLAN-POSN(INDEX)
      IF (DT.LT.100.) GO TO 560
      CALL JBITM (ICAR(INDEX),36,4,13)
      GO TO 590
  560 CONTINUE
      IACC=IBITM(ICAR(INDEX),36,4)
      IF (CURB(LINK)-POSN(INDEX).GT.300..AND.LLAN(LINK,LANNEW).EQ.LLAN(L
     1INK,LANE)) GO TO 590
      LA=LANCAR(LINK,LANNEW)
      IF (LA.EQ.0) GO TO 570
      ILST=ILSTCAR(LINK,LANNEW)
      DT=POSN(ILST)-POSN(INDEX)
      IF (DT.GE.100.) GO TO 570
      IF (DT.LE.0.) DT=1.
      IF ((VEL(ILST)*VEL(ILST)-VEL(INDEX)*VEL(INDEX))/(2.*DT).GE.-12.) G
     1O TO 570
      ACCEL(INDEX)=(VEL(ILST)-VEL(INDEX))/DT*A0
      IF (IACC.EQ.5.OR.IACC.EQ.6) GO TO 580
      CALL JBITM (ICAR(INDEX),36,4,5)
      GO TO 590
  570 CALL JBITM (ICAR(INDEX),36,4,13)
      GO TO 590
  580 IF (IACC.EQ.6.AND.ACCEL(INDEX).GT.0.) CALL JBITM (ICAR(INDEX),36,4
     1,5)
  590 CONTINUE
      INDEX=NXTCAR(INDEX)
      IF (INDEX.GT.0) GO TO 100
  600 CONTINUE
  610 CONTINUE
      RETURN
C
      END


      SUBROUTINE GENBHR (LINK,LANE,NUM,IFLAG)
      COMMON /A1/ IREM,INEXT,IMAX,ISIZE,IFRSTCR(84,3),ILSTCAR(84,3),VEL(
     11500),POSN(1500),ACCEL(1500),ICAR(1500),NXTCAR(1500),IPRECAR(1500)
      COMMON /PARAM/ A0,FREEK,REACT,CYCLE
      COMMON /RELAT/ LINCAR(84),NOLINK,LANCAR(84,3),NOLANE(84)
      COMMON /PV/ P(84,3),V(84,3)
      COMMON /GEOM/ CURB(84),DBNDRY(84,6)
      COMMON /TURN/ NREC(84,6),LREC(84,6),PTURN(84,6),PTR(84,2)
      COMMON /CAR/ IACTYP,TVEL,ITURN,EL,JACTYP
      COMMON /KWNTZ/ NCAR(84),NOUTPT,ISTART,IEND,KGEN(84)
      COMMON /STARTS/ T2,MGO
      COMMON /MSK/ MASK(20)
      REAL NORMAL
      IBITM(IX,M,N)=ISHFTR(IX,M).AND.MASK(N)
      LNUM=IPRECAR(NUM)
      MAXDAC=IBITM(ICAR(NUM),50,4)
```

TABLE B-2, cont.

```
      LACTYP=IFLAG
      IFLAG=0
      MGO=0
      ACC1=ACCEL(NUM)
      IF (NUM.EQ.IFRSTCR(LINK,LANE)) GO TO 190
      DIST=P(LINK,LANE)-POSN(NUM)
      IF (DIST.LE.0.) DIST=.001
      IF (DIST.GT.200.) GO TO 190
      IF (JACTYP.EQ.1.AND.LACTYP.NE.1) GO TO 100
      GO TO 110
  100 IF (LACTYP.EQ.0) GO TO 110
      IF (ACCEL(LNUM).EQ.0.) GO TO 110
      IF (IACTYP.NE.1.AND.IACTYP.NE.6) GO TO 110
      T2=CYCLE-VEL(LNUM)/ACCEL(LNUM)+1.
      IF (T2.GE.CYCLE) GO TO 110
      V2=V(LINK,LANE)+ACCEL(LNUM)
      P2=P(LINK,LANE)+.5*ACCEL(LNUM)
      ACCFOL=A0*(V2-VEL(NUM))/(P2-POSN(NUM))
      MGO=1
      GO TO 120
  110 CONTINUE
      ACCFOL=(V(LINK,LANE)-VEL(NUM))*A0/DIST
      IF (ACCFOL.LE.0.) GO TO 140
  120 CONTINUE
      ACCFRE=FREEK*(TVEL-VEL(NUM))
      IF (ACCFRE.LE.ACCFOL) GO TO 200
      ACCEL(NUM)=ACCFOL
      IF (ACCFOL.LE.0..AND.VEL(NUM).LE.0.) GO TO 190
      IF (ACCFOL.GT.FLOAT(MAXDAC)) ACCEL(NUM)=FLOAT(MAXDAC)
  130 IF (IACTYP.EQ.8) RETURN
      IFLAG=8
      GO TO 230
  140 CONTINUE
      IF (JACTYP.GT.6.AND.JACTYP.LT.12) GO TO 120
      IF (IACTYP.EQ.1) GO TO 180
      IF (LACTYP.GT.6.AND.LACTYP.LT.12) GO TO 120
      IF (ACCEL(LNUM).EQ.0.) GO TO 150
      IF (VEL(LNUM).LT.0.) VEL(LNUM)=0.
      S1F=-(VEL(LNUM)*VEL(LNUM))/(2.*ACCEL(LNUM))+POSN(LNUM)
      GO TO 160
  150 S1F=VEL(LNUM)*CYCLE+POSN(LNUM)
  160 EL=FLOAT(IBITM(ICAR(LNUM),40,6))
      S2F=S1F-EL
      V22=VEL(NUM)+ACC1*REACT
      IF (V22.LE.0.) V22=.001
      S22=POSN(NUM)+VEL(NUM)*REACT+.5*ACCEL(NUM)*REACT*REACT
      XI=S2F-S22
      IF (XI.LE.0.) XI=1.
      ACCEL(NUM)=-(V22*V22)/(2.*XI)
      IF (ACCEL(NUM).LT.-FLOAT(IBITM(ICAR(NUM),46,4))) GO TO 170
      S22=S22+CYCLE*V22
      XI=S2F-S22
      IF (XI.LT.0.) GO TO 170
      IF (XI.EQ.0.) XI=1
      ACC=-(V22*V22)/(2.*XI)
      IF (ACC.LT.-10.) GO TO 170
      ACCEL(NUM)=0.
      IF (IACTYP.EQ.8) RETURN
      IFLAG=8
      GO TO 230
  170 IF (IACTYP.EQ.2) RETURN
```

TABLE B-2, cont.

```
      IFLAG=2
      GO TO 230
  180 IF (IACTYP.NE.1) IFLAG=1
      ACCEL(NUM)=0.
      GO TO 230
  190 CONTINUE
      ACCFRE=FREEK*(TVEL-VEL(NUM))
  200 ACCEL(NUM)=FLOAT(MAXDAC)
      IF (ACCFRE.LE.ACCEL(NUM)) ACCEL(NUM)=ACCFRE
      IF (IACTYP.NE.7) IFLAG=7
      IF (NUM.EQ.IFRSTCR(LINK)) GO TO 230
      IF (LINK.LE.NOUTPT) GO TO 230
      POSNEX=ACCEL(NUM)*CYCLE*CYCLE/2.+VEL(NUM)*CYCLE+POSN(NUM)
      IF (POSNEX.LE.DBNDRY(LINK,ITURN)) GO TO 230
      II=NPEC(LINK,ITURN)
      JJ=LPEC(LINK,ITURN)
      IF (II.LE.0) GO TO 230
      KK=ILSTCAR(II,JJ)
      IF (KK.EQ.0) GO TO 230
      IF (POSN(KK)+DBNDRY(LINK,ITURN)-POSN(NUM).GT.300.) GO TO 230
      IF (II.GT.LINK) GO TO 220
      XI=DBNDRY(LINK,ITURN)-POSN(NUM)+P(II,JJ)
      IF (XI.LE.0.) XI=1.
      ACCFOL=A0*(V(II,JJ)-VEL(NUM))/XI
  210 IF (ACCEL(NUM).LE.ACCFOL) GO TO 230
      ACCEL(NUM)=ACCFOL
      GO TO 130
  220 XI=DBNDRY(LINK,ITURN)-POSN(NUM)+POSN(KK)
      IF (XI.LE.0.) XI=1.
      ACCFOL=A0*(VEL(KK)-VEL(NUM))/XI
      GO TO 210
  230 CONTINUE
      RETURN
C
      END
```

```
      SUBROUTINE REDAPP (LINK,LANE,NUM,ACCTYP)
      COMMON /A1/ IREM,INEXT,IMAX,ISIZE,IFRSTCR(84,3),ILSTCAR(84,3),VEL(
     11500),POSN(1500),ACCEL(1500),ICAR(1500),NXTCAR(1500),IPRECAR(1500)
      COMMON /PARAM/ A0,FREEK,REACT,CYCLE
      COMMON /GEOM/ CURB(84),DBNDRY(84,6)
      COMMON /PV/ P(84,3),V(84,3)
      COMMON /TM/ ITIME
      COMMON /CAR/ IACC,TVEL,ITURN,EL,IACCL
      COMMON /MSK/ MASK(20)
      INTEGER ACCTYP
      REAL NORMAL
      IBITM(IX,M,N)=ISHFTR(IX,M).AND.MASK(N)
      LNUM=IPRECAR(NUM)
      N=LNUM
      IFLAG=0
      LACC=ACCTYP
      ACCTYP=0
      ACC1=ACCEL(NUM)
      IF (POSN(NUM).GE.CURB(LINK)-400.) GO TO 140
  100 CALL GENBHR (LINK,LANE,NUM,LACC)
      ACCTYP=LACC
      GO TO 480
  110 GO TO (120,120,100,100,130,130), ITURN
```

TABLE B-2, cont

```
 120 CALL RIGHT (LINK,LANE,NUM,LACC)
     ACCTYP=LACC
     RETURN
 130 CALL LEFT (LINK,LANE,NUM,ITIME,LACC)
     ACCTYP=LACC
     RETURN
 140 CONTINUE
     IF (NUM.NE.IFRSTCR(LINK,LANE)) GO TO 250
     IF (IACC.NE.1) GO TO 190
     X=CURB(LINK)-15.
     IF (POSN(NUM).LT.X) GO TO 150
     ACCEL(NUM)=0.
     VEL(NUM)=0.
     IF (IACC.EQ.1) RETURN
     ACCTYP=1
     RETURN
 150 CONTINUE
     ACCDES=FLOAT(IBITM(ICAR(NUM),50,4))
     ACCEL(NUM)=FREEK*(TVEL-VEL(NUM))
 160 CONTINUE
     ACC=ACCDES/2.
     IF (ACCEL(NUM).LE.ACC) GO TO 170
     ACCEL(NUM)=ACC
 170 CONTINUE
     IF (IACC.EQ.7) GO TO 180
     ACCTYP=7
 180 CONTINUE
     RETURN
 190 CONTINUE
     IF (IACC.NE.12) GO TO 200
     IF (VEL(NUM).GT.0.) GO TO 220
     VEL(NUM)=0.
     IF (IACC.EQ.1) RETURN
     ACCTYP=1
     RETURN
 200 CONTINUE
     IF (POSN(NUM).GT.CURB(LINK)+3.) GO TO 110
     IF (CURB(LINK)-POSN(NUM).LT.300.) GO TO 210
     CALL GENBHP (LINK,LANE,NUM,ACCTYP)
     RETURN
 210 CONTINUE
     V2=VEL(NUM)+ACC1*REACT
     SF=CURB(LINK)-1.
     S2=POSN(NUM)+VEL(NUM)*REACT+.5*ACC1*REACT*REACT
     DELS=SF-S2
     IF (DELS.LE.0.) DELS=1.
     IF (V2.LE.0.) V2=.001
     ACCEL(NUM)=-(V2*V2)/(2.*DELS)
     IF (ACCEL(NUM).GT.-FLOAT(IBITM(ICAR(NUM),46,4))) GO TO 240
 220 CONTINUE
     IF (IFLAG.EQ.5) GO TO 270
 230 IF (IACC.EQ.12) RETURN
     ACCTYP=12
     RETURN
 240 CONTINUE
     DELS=SF-S2-V2*CYCLE
     IF (DELS.LE.0.) GO TO 220
     A=-(V2*V2)/(2.*DELS)
     IF (A.LT.-10.) GO TO 220
     ACCEL(NUM)=0.
     IF (IFLAG.EQ.5) GO TO 250
```

TABLE B-2, cont.

```
      IF (IACC.EQ.7) RETURN
      ACCTYP=7
      RETURN
  250 DELS=P(LINK,LANE)-POSN(NUM)
      IF (DELS.LE.0.) DELS=1.
      ACC=A0*(V(LINK,LANE)-VEL(NUM))/DELS
      IF (ACC.GE.ACCEL(NUM)) GO TO 260
      ACCEL(NUM)=ACC
      IF (IACC.NE.8) ACCTYP=8
      RETURN
  260 IF (IACC.NE.7) ACCTYP=7
      RETURN
  270 DELS=P(LINK,LANE)-POSN(NUM)
      IF (DELS.LE.0.) DELS=1.
      ACC=A0*(V(LINK,LANE)-VEL(NUM))/DELS
      IF (ACC.GE.ACCEL(NUM)) GO TO 230
      ACCEL(NUM)=ACC
      IF (IACC.NE.8) ACCTYP=8
      RETURN
  280 CONTINUE
      N=IPRECAR(NUM)
      X=P(LINK,LANE)-POSN(NUM)
      IF (X.GT.300.) GO TO 150
      IF (IACC.EQ.12) GO TO 410
      IF (IACC.EQ.1) GO TO 430
  290 IF (POSN(NUM).GT.CURB(LINK)) GO TO 110
      IF (POSN(N).LE.CURB(LINK)) GO TO 300
      IFLAG=5
      GO TO 210
  300 CONTINUE
      IF (LACC.LT.7.OR.LACC.EQ.12.AND.X.LT.200.) GO TO 360
      IF (IBITM(ICAR(N),33,3).NE.ITURN) GO TO 210
      DELS=P(LINK,LANE)-POSN(NUM)
      IF (DELS.LE.0.) DELS=1.
      ACCEL(NUM)=A0*(V(LINK,LANE)-VEL(NUM))/DELS
      IF (ACCEL(NUM).LE.0.) GO TO 350
  310 CONTINUE
      ACC=FREEK*(TVEL-VEL(NUM))
      IF (ACCEL(NUM).LT.ACC) GO TO 320
      ACCDES=FLOAT(IBITM(ICAR(NUM),50,4))
      GO TO 160
  320 CONTINUE
      ACC=FLOAT(IBITM(ICAR(NUM),50,4)/2)
      IF (ACCEL(NUM).LE.ACC) GO TO 330
      ACCEL(NUM)=ACC
      ACCEL(NUM)=0.
      GO TO 330
      IF (ACCEL(N).EQ.0.) GO TO 370
  330 CONTINUE
      IF (IACC.EQ.8) GO TO 340
      ACCTYP=8
  340 CONTINUE
      RETURN
  350 CONTINUE
      IF (IACCL.EQ.12.OR.IACCL.EQ.2) GO TO 360
      IF (IACCL.NE.1) GO TO 310
  360 CONTINUE
      IF (ACCEL(N).EQ.0.) GO TO 370
      S1F=-(VEL(N)*VEL(N))/(2.*ACCEL(N))+POSN(N)
      GO TO 380
  370 S1F=VEL(N)*CYCLE+POSN(N)
```

TABLE B-2, cont.

```
380 EL=FLOAT(IBITM(ICAR(N),40,6))
    S2F=S1F-EL
    V22=VEL(NUM)+ACC1*REACT
    IF (V22.LE.0.) V22=.001
    IF (ACC1.EQ.0.) GO TO 390
    S22=POSN(NUM)+VEL(NUM)*REACT+.5*ACC1*REACT*REACT
    GO TO 400
390 CONTINUE
    S22=POSN(NUM)+VEL(NUM)*REACT
400 XIX=S2F-S22
    IF (XIX.LE.0.) XIX=1.
    ACCEL(NUM)=-(V22*V22)/(2.*XIX)
    ACC=-FLOAT(IBITM(ICAR(NUM),46,4))
    IF (ACCEL(NUM).LT.ACC) GO TO 420
    DELS=S2F-S22-V22*CYCLE
    IF (DELS.LT.0.) GO TO 420
    IF (DELS.EQ.0.) DELS=1.
    A=-V22*V22/(2.*DELS)
    IF (A.LT.-10.) GO TO 420
    ACCEL(NUM)=0.
    IF (IACC.EQ.8) RETURN
    ACCTYP=8
    RETURN
410 CONTINUE
    IF (IACCL.NE.12) GO TO 290
    ACCEL(NUM)=ACC1
420 CONTINUE
    IF (VEL(NUM).LE.0.) GO TO 430
    IF (IACC.EQ.12) RETURN
    ACCTYP=12
    RETURN
430 ELL=FLOAT(IBITM(ICAR(N),40,6))
    IF (POSN(NUM).GT.POSN(N)-ELL-10.) GO TO 470
    ACC=FREEK*(TVEL-VEL(NUM))
    P22=POSN(NUM)+.5*ACC*CYCLE*CYCLE
    V22=VEL(NUM)+ACC*CYCLE
    P12=POSN(N)+VEL(N)*REACT+.5*ACCEL(N)*REACT*REACT
    V12=VEL(N)+ACCEL(N)*REACT
    IF (P22.GE.P12-ELL) GO TO 470
    IF (ACCEL(N).LT.0.) GO TO 450
    P1F=P12+V12*CYCLE+.5*ACCEL(N)*CYCLE*CYCLE
440 P2F=P1F-ELL
    IF (P2F.LE.P22) GO TO 470
    A=-V22*V22/(2.*(P2F-P22))
    GO TO 460
450 P1F=P12-V12*V12/(2.*ACCEL(N))
    GO TO 440
460 IF (A.LT.-12.) GO TO 470
    ACCEL(NUM)=ACC
    IF (IACC.NE.7) ACCTYP=7
    RETURN
470 CONTINUE
    ACCEL(NUM)=0.
    VEL(NUM)=0.
    IF (IACC.EQ.1) RETURN
    ACCTYP=1
480 CONTINUE
    RETURN
C
    END
```

TABLE B-2, cont.

```
      SUBROUTINE RIGHT (LINK,LANE,NUM,IEQN)
      COMMON /A1/ IREM,INEXT,IMAX,ISIZE,IFRSTCR(84,3),ILSTCAR(84,3),VEL(
     11500),POSN(1500),ACCEL(1500),ICAR(1500),NXTCAR(1500),IPRECAR(1500)
      COMMON /PARAM/ A0,FREEK,REACT,CYCLE
      COMMON /GEOMET/ RADIUS(84,6),BEGINPT(84,6)
      COMMON /PV/ P(84,3),V(84,3)
      COMMON /TURN/ NREC(84,6),LREC(84,6),PTURN(84,6),PTR(84,2)
      COMMON /RELAT/ LINCAR(84),NOLINK,LANCAR(84,3),NOLANE(84)
      COMMON /GEOM/ CURB(84),DPNDRY(84,6)
      COMMON /CAR/ IACC,TVEL,ITURN,EL,IACCL
      COMMON /MSK/ MASK(20)
      EQUIVALENCE (VTURN,ABSVT)
      IBITH(IX,M,N)=ISHFTR(IX,M).AND.MASK(N)
      IFLAG=IEQN
      IEQN=0
      NT=1
      IF (ITURN.EQ.2) NT=2
      ACCPR=ACCEL(NUM)
      IF (POSN(NUM).GE.CURB(LINK)-300.) GO TO 110
  100 CALL GENBHR (LINK,LANE,NUM,IFLAG)
      IEQN=IFLAG
      RETURN
  110 CONTINUE
      IF (NUM.EQ.IFRSTCR(LINK,LANE)) GO TO 120
      IF (IACCL.EQ.1.AND.IFLAG.NE.1) GO TO 100
      IF (IACC.EQ.1.AND.IACCL.NE.1) GO TO 100
  120 CONTINUE
      VTURN=SQRT(9.66*RADIUS(LINK,NT))
      DECMIN=-FLOAT(IBITH(ICAR(NUM),46,4))
      X=BEGINPT(LINK,NT)-(POSN(NUM)+VEL(NUM)*REACT+.5*ACCEL(NUM)*REACT*R
     1EACT)
      IF (X.LE.0.) GO TO 150
      CALL GENBHR (LINK,LANE,NUM,IFLAG)
      IEQN=IFLAG
      ABSVX=VEL(NUM)+ACCPR*REACT
      IF (ABSVX.LT.0.) ABSVX=0.
      ACC=(ABSVT*ABSVT-ABSVX*ABSVX)/(2.*X)
      IF (ACC.GT.ACCEL(NUM)) GO TO 140
      IF (ACC.GT.DECMIN.AND.ACC.LT.0.) GO TO 130
      GO TO 140
  130 XX=CURB(LINK)-(POSN(NUM)+VEL(NUM)*REACT+.5*ACCPR*REACT*REACT+(VEL(
     1NUM)+ACCPR*REACT)*CYCLE+.5*ACCEL(NUM)*CYCLE*CYCLE)
      ABSV=VEL(NUM)+ACCPR*REACT+ACCEL(NUM)*CYCLE
      IF (ABSV.LT.0.) ABSV=0.
      A=(ABSVT*ABSVT-ABSV*ABSV)/(2.*XX)
      IF (A.GT.-10.) GO TO 140
      ACCEL(NUM)=ACC
      IEQN=11
  140 CONTINUE
      IF (NUM.EQ.IFRSTCR(LINK,LANE)) GO TO 160
      N=IPRECAR(NUM)
      IF (IBITH(ICAR(N),33,3).NE.ITURN) GO TO 160
      GO TO 190
  150 ACCEL(NUM)=(VTURN-(VEL(NUM)+ACCPR*REACT))/CYCLE
      IF (IACC.NE.10) IEQN=10
      GO TO 220
  160 CONTINUE
      X=POSN(NUM)+VEL(NUM)*REACT+.5*ACCPR*REACT*REACT
      IF (X.LT.BEGINPT(LINK,NT)) GO TO 190
```

TABLE B-2, cont.

```
      N=NREC(LINK,ITURN)
      L=LREC(LINK,ITURN)
      NO=LANCAR(N,L)
      IF (NO.EQ.0) GO TO 190
      NO=ILSTCAR(N,L)
      IF (N.LT.LINK) GO TO 170
      X=DBNDRY(LINK,ITURN)-POSN(NUM)+POSN(NO)
      IF (X.LE.0.) X=1.
      ACC=A0*(VEL(NO)-VEL(NUM))/X
      GO TO 180
  170 CONTINUE
      X=DBNDRY(LINK,ITURN)-POSN(NUM)+P(N,L)
      IF (X.LE.0.) X=1.
      ACC=A0*(V(N,L)-VEL(NUM))/X
  180 CONTINUE
      IF (ACCEL(NUM).LE.ACC) GO TO 190
      IF (ACC.GT.DECMIN) GO TO 190
      ACCEL(NUM)=ACC
      IF (IACC.NE.8) IEQN=8
      GO TO 220
  190 CONTINUE
      CALL CONST (LINK,LANE,NUM,ICON,1)
      IF (ICON.EQ.0) GO TO 220
      X=CURB(LINK)-POSN(NUM)
      IF (X.LE.0.) GO TO 200
      IF (X.GT.50.) GO TO 220
      IF (X.EQ.0.) X=.001
      V2=VEL(NUM)+ACCPR*REACT
      IF (V2.LE.0.) V2=0.
      ACC=-(V2*V2)/X
      GO TO 210
  200 ACC=-2.*FLOAT(IBITM(ICAR(NUM),46,4))
  210 CONTINUE
      IF (ACC.LT.ACCEL(NUM)) ACCEL(NUM)=ACC
      IF (IACC.NE.2) IEQN=2
  220 CONTINUE
      ACCMAX=FLOAT(IBITM(ICAR(NUM),50,4))
      IF (ACCEL(NUM).GT.ACCMAX) ACCEL(NUM)=ACCMAX
      RETURN
C
      END


      SUBROUTINE LEFT (LINK,LANE,NUM,ITIME,IEQN)
      COMMON /A1/ IREM,INEXT,IMAX,ISIZE,IFRSTCR(84,3),ILSTCAR(84,3),VEL(
     11500),POSN(1500),ACCEL(1500),ICAR(1500),NXTCAR(1500),IPRECAR(1500)
      COMMON /PARAM/ A0,FREEK,REACT,CYCLE
      COMMON /OPP/ IOPP(84,5)
      COMMON /RELAT/ LINCAR(84),NOLINK,LANCAR(84,3),NOLANE(84)
      COMMON /GEOMET/ RADIUS(84,6),BEGINPT(84,6)
      COMMON /GEOM/ CURB(84),DBNDRY(84,6)
      COMMON /TURN/ NREC(84,6),LREC(84,6),PTURN(84,6),PTR(84,2)
      COMMON /PV/ P(84,3),V(84,3)
      COMMON /CAR/ IACC,TVEL,ITURN,EL,IACCL
      COMMON /MSK/ MASK(20)
      IBITM(IX,M,N)=ISHFTR(IX,M).AND.MASK(N)
      IEGO=IEQN
      IEQN=0
      NTD=5
      NTF=6
```

TABLE B-2, cont.

```
      IF (ITURN.EQ.6) GO TO 100
      NTD=3
      NTF=4
  100 CONTINUE
      ACCPR=ACCEL(NUM)
      IF (BEGINPT(LINK,NTF)-POSN(NUM).LT.300.) GO TO 110
      CALL GENBHR (LINK,LANE,NUM,IFGO)
      IEQN=IFGO
      GO TO 540
  110 CONTINUE
      JOPP=IOPP(LINK,3)
      IF (IOPP.NE.0) GO TO 120
      IA=3
      GO TO 190
  120 CONTINUE
      NOLLL=NOLANE(JOPP)
      JA=0
      DO 180 LANOP=1,NOLLL
      CALL SIGNAL (JSIG,JOPP,LANOP,ITIME)
      IF (JSIG.EQ.1.OR.JSIG.EQ.2.OR.JSIG.EQ.4.OR.JSIG.EQ.6.OR.JSIG.EQ.12
     1.OR.JSIG.EQ.14) GO TO 130
      IA=2
      GO TO 170
  130 CONTINUE
      IF (JSIG.EQ.14.OR.JSIG.EQ.2) GO TO 140
      IA=3
      GO TO 170
  140 CONTINUE
      CALL CONST (LINK,LANE,NUM,ICON,0)
      IF (ICON.EQ.1) GO TO 150
      CALL CONLT (LINK,LANE,NUM,ICON)
  150 CONTINUE
      IF (ICON.EQ.0) GO TO 160
      IA=1
      GO TO 170
  160 CONTINUE
      IA=2
  170 CONTINUE
      IF (IA.EQ.1) GO TO 190
      IF (IA.EQ.2) JA=2
  180 CONTINUE
      IF (IA.EQ.3.AND.JA.EQ.2) IA=2
  190 CONTINUE
      GO TO (370,440,230,270,470,470,470,470,200,220,320,440), IACC
  200 CONTINUE
      IF (IACC.NE.9) IEQN=9
      V2=VEL(NUM)+ACCPR*REACT
      IF (V2.LT.0.) V2=0.
      ACCEL(NUM)=(SQRT(9.56*RADIUS(LINK,NTD))-V2)/CYCLE
  210 IF (NUM.EQ.IFRSTCR(LINK,LANE)) GO TO 500
      N=IPRECAR(NUM)
      IF (IBITH(ICAR(N),33,3).NE.ITURN) GO TO 500
      XX=P(LINK,LANE)-POSN(NUM)
      IF (XX.LE.0.) XX=.001
      ACC=A0*(V(LINK,LANE)-VEL(NUM))/XX
      IF (ACC.LT.ACCEL(NUM)) ACCEL(NUM)=ACC
      GO TO 540
  220 CONTINUE
      IF (IACC.NE.10) IEQN=10
      V2=VEL(NUM)+ACCPR*REACT
      IF (V2.LT.0.) V2=0.
```

TABLE B-2, cont.

```
      ACCEL(NUM)=(SQRT(9.66*RADIUS(LINK,NTF))-V2)/CYCLE
      GO TO 210
  230 CONTINUE
      GO TO (240,250,260), IA
  240 CONTINUE
      ACCEL(NUM)=0.
      IF (IACC.NE.3) IEQN=3
      GO TO 490
  250 CONTINUE
      CALL GAP (ICON,LINK,LANE,ITIME)
      GO TO (260,240), ICON
  260 CALL GENBHR (LINK,LANE,NUM,IFGO)
      IEQN=9
      GO TO 530
  270 CONTINUE
      GO TO (280,300,310), IA
  280 CONTINUE
      IF (VEL(NUM).LE.0.) GO TO 290
      IF (IACC.NE.4) IEQN=4
      GO TO 490
  290 CONTINUE
      ACCEL(NUM)=0.
      VEL(NUM)=0.
      IF (IACC.NE.3) IEQN=3
      GO TO 490
  300 CONTINUE
      CALL GAP (ICON,LINK,LANE,ITIME)
      GO TO (310,280), ICON
  310 IF ((POSN(NUM)+VEL(NUM)*REACT+.5*ACCPR*REACT*REACT).GT.BEGINPT(LIN
     1K,NTF)) GO TO 200
      GO TO 220
  320 CONTINUE
      IF (POSN(NUM)+VEL(NUM)*REACT+.5*ACCPR*REACT*REACT+30.*CYCLE.GT.BEG
     1INPT(LINK,NTF)) GO TO 330
      GO TO 530
  330 GO TO (340,350,360), IA
  340 X=BEGINPT(LINK,NTD)+(DBNDRY(LINK,ITURN)-BEGINPT(LINK,NTD))/4.-(POS
     1N(NUM)+VEL(NUM)*REACT+.5*ACCPR*REACT*REACT)
      IF (X.LE.0.) X=.001
      V2=VEL(NUM)+ACCPR*REACT
      IF (V2.LE.0.) V2=0.
      ACCEL(NUM)=-(V2*V2)/(2.*X)
      IEQN=4
      GO TO 540
  350 CALL GAP (ICON,LINK,LANE,ITIME)
      GO TO (360,340), ICON
  360 IF (POSN(NUM)+VEL(NUM)*REACT+.5*ACCPR*REACT*REACT.GT.BEGINPT(LINK,
     1NTF)) GO TO 220
      GO TO 490
  370 IF (POSN(NUM)+VEL(NUM)*REACT+.5*ACCPR*REACT*REACT+30.*CYCLE.GT.BEG
     1INPT(LINK,NTF)) GO TO 390
  380 CALL GENBHR (LINK,LANE,NUM,IFGO)
      IEQN=IFGO
      GO TO 540
  390 GO TO (400,410,420), IA
  400 IF (NUM.NE.IFRSTCR(LINK,LANE)) GO TO 380
      IEQN=3
      GO TO 540
  410 IF (POSN(NUM).LT.BEGINPT(LINK,NTF)-100.) GO TO 430
      CALL GAP (ICON,LINK,LANE,ITIME)
      GO TO (420,400), ICON
```

TABLE B-2, cont.

```
420 CALL CONST (LINK,LANE,NUM,ICON,0)
    IF (ICON.EQ.1) GO TO 400
430 CALL GENBHR (LINK,LANE,NUM,IFGO)
    IFON=IFGO
    GO TO 540
440 IF (POSN(NUM)+VEL(NUM)*REACT+.5*ACCPR*REACT*REACT+30.*CYCLE.LT.BEG
   1INPT(LINK,NTF)) GO TO 450
    IFON=11
    XX=BEGINPT(LINK,NTF)-(POSN(NUM)+VEL(NUM)*REACT+.5*ACCPR*REACT*REAC
   1T)
    IF (XX.LE.0.) GO TO 330
    V2=VEL(NUM)+ACCPR*REACT
    IF (V2.LT.0.) V2=0.
    ACCEL(NUM)=(9.66*RADIUS(LINK,NTF)-V2*V2)/(2.*XX)
    GO TO 330
450 CALL GENBHR (LINK,LANE,NUM,IFGO)
    IFON=IFGO
    XX=BEGINPT(LINK,NTF)-(POSN(NUM)+VEL(NUM)*REACT+.5*ACCPR*REACT*REAC
   1T)
    IF (XX.LE.0.) XX=1.
    V2=VEL(NUM)+ACCPR*REACT
    IF (V2.LT.0.) V2=0.
    ACC=(9.66*RADIUS(LINK,NTF)-V2*V2)/(2.*XX)
    IF (ACC.GE.ACCEL(NUM)) GO TO 540
    IF (ACC.GE.-FLOAT(IBITM(ICAR(NUM),46,4)).AND.ACC.LT.0.) GO TO 460
    ACCEL(NUM)=ACC
    IFON=11
    GO TO 540
460 XX=BEGINPT(LINK,NTF)-(POSN(NUM)+VEL(NUM)*(CYCLE+REACT))
    IF (XX.LT.0.) GO TO 540
    IF (XX.LE.0.) XX=.101
    A=(9.66*RADIUS(LINK,NTF)-VEL(NUM)*VEL(NUM))/(2.*XX)
    IF (A.LE.-10.) GO TO 540
    ACCEL(NUM)=ACC
    IFON=11
    GO TO 540
470 XX=BEGINPT(LINK,NTF)-(POSN(NUM)+VEL(NUM)*REACT+.5*ACCPR*REACT*REAC
   1T)
    IF (XX.GT.30.*CYCLE) GO TO 450
    IF (XX.LE.0.) GO TO 220
    GO TO (340,480,220), IA
480 CALL GAP (ICON,LINK,LANE,ITIME)
    GO TO (220,340), ICON
490 CONTINUE
    IF (NUM.EQ.IFRSTCR(LINK,LANE)) GO TO 500
    N=IPRECAR(NUM)
    IF (IBITM(ICAR(N),33,3).NE.ITURN.AND.POSN(NUM).GE.BEGINPT(LINK,NTF
   1)) GO TO 500
    GO TO 530
500 CONTINUE
    X=POSN(NUM)+VEL(NUM)*REACT+.5*ACCEL(NUM)*REACT*REACT
    IF (X.LT.DBNDRY(LINK,ITURN)) GO TO 530
    N=NFFC(LINK,ITURN)
    L=LFFC(LINK,ITURN)
    NO=LANCAR(N,L)
    IF (NO.EQ.0) GO TO 530
    NO=ILSTCAR(N,L)
    IF (N.LT.LINK) GO TO 510
    XX=DBNDRY(LINK,ITURN)-POSN(NUM)+POSN(NO)
    IF (XX.LE.0.) XX=1.
    ACC=A0*(VEL(NO)-VEL(NUM))/XX
```

TABLE B-2, cont.

```
      GO TO 520
  510 CONTINUE
      XX=DPNDRY(LINK,ITURN)-POSN(NUM)+P(N,L)
      IF (XX.EQ.0.) XX=1.
      ACC=A0*(V(N,L)-VEL(NUM))/XX
  520 CONTINUE
      IF (ACCEL(NUM).LE.ACC) GO TO 530
      ACCEL(NUM)=ACC
      IF (IACC.NE.8) IEQN=8
  530 CONTINUE
      ACCMAX=FLOAT(IBITM(ICAR(NUM),50,4))
      IF (ACCEL(NUM).GT.ACCMAX) ACCEL(NUM)=ACCMAX
  540 CONTINUE
      RETURN
C
      END


      SUBROUTINE CONLT (LINK,LANE,NUM,ICON)
      COMMON /A1/ IREM,INEXT,IMAX,ISIZE,IFRSTCR(84,3),ILSTCAR(84,3),VEL(
     11500),POSN(1500),ACCEL(1500),ICAR(1500),NXTCAR(1500),IPRECAR(1500)
      COMMON /GEOM/ CURB(84),DPNDRY(84,6)
      COMMON /OPP/ IOPP(84,5)
      COMMON /RELAT/ LINCAR(84),NOLINK,LANCAR(84,3),NOLANE(84)
      I=IOPP(LINK,3)
      IF (I.EQ.0) GO TO 130
      NOLLL=NOLANE(I)
      DO 120 LAN=1,NOLLL
      IF (LANCAR(I,LAN).EQ.0) GO TO 110
      NO=IFRSTCR(I,LAN)
  100 IF (NO.EQ.0) GO TO 120
      IF (POSN(NO).LE.CURB(I)) GO TO 120
      IF (POSN(NO).LE.CURB(I)+35.) GO TO 140
      NO=NXTCAR(NO)
      GO TO 100
  110 CONTINUE
  120 CONTINUE
  130 CONTINUE
      ICON=0
C
C     ICON=0 MEANS NO CONFLICT EXISTS
C
      RETURN
  140 CONTINUE
      ICON=1
C
C     ICON=1 MEANS CONFLICT EXISTS
C
      RETURN
C
      END


      SUBROUTINE CONST (LINK,LANE,NUM,ICON,IFL)
      COMMON /A1/ IREM,INEXT,IMAX,ISIZE,IFRSTCR(84,3),ILSTCAR(84,3),VEL(
     11500),POSN(1500),ACCEL(1500),ICAR(1500),NXTCAR(1500),IPRECAR(1500)
      COMMON /RELAT/ LINCAR(84),NOLINK,LANCAR(84,3),NOLANE(84)
      COMMON /OPP/ IOPP(84,5)
      COMMON /GEOM/ CURB(84),DPNDRY(84,6)
```

TABLE B-2, cont.

```
      COMMON /CAP/ IACC,TVEL,ITURN,EL,IACCL
      COMMON /MSK/ MASK(20)
      IBITM(IX,M,N)=ISHFTR(IX,M).AND.MASK(N)
      IF (IFL.EQ.1) GO TO 150
      I=ICFP(LINK,1)
  100 IF (I.EQ.0) GO TO 130
      NOLLL=NOLANE(I)
      DO 140 LAN=1,NOLLL
      IF (LANCAR(I,LAN).EQ.0) GO TO 140
      NO=IFRSTCR(I,LAN)
  110 IF (NO.EQ.0) GO TO 140
      IF (POSN(NO).LE.CURB(I)) GO TO 140
      IF (VEL(NO).LE.0.) GO TO 140
      IF (POSN(NO).GT.CURB(I)+35.) GO TO 120
      ICON=1
      RETURN
  120 CONTINUE
      NO=NXTCAR(NO)
      GO TO 110
  130 CONTINUE
  140 CONTINUE
      IF (I.EQ.IOPP(LINK,2)) GO TO 150
      I=IOFP(LINK,2)
      GO TO 100
  150 I=ICFP(LINK,4)
  160 IF (I.EQ.0) GO TO 190
      NOLLL=NOLANE(I)
      DO 200 LAN=1,NOLLL
      IF (LANCAR(I,LAN).EQ.0) GO TO 200
      NO=IFPSTCR(I,LAN)
  170 IF (NO.EQ.0) GO TO 200
      IF (POSN(NO).LT.CURB(I)) GO TO 200
      IF (VEL(NO).LE.0.) GO TO 200
      IF (POSN(NO).LT.CURB(I)+55.) GO TO 180
      NO=NXTCAR(NO)
      GO TO 170
  180 CONTINUE
      IF (IBITM(ICAR(NO),33,3).GE.3.AND.ITURN.LT.5) GO TO 220
  190 CONTINUE
  200 CONTINUE
      IF (I.EQ.IOPP(LINK,5)) GO TO 210
      I=ICFP(LINK,5)
      GO TO 160
  210 CONTINUE
      ICON=0
C
C     ICON=0 MEANS NO CONFLICT EXISTS
C
      RETURN
  220 CONTINUE
      ICON=1
C
C     ICON=1 MEANS CONFLICT EXISTS
C
      RETURN
C
      END


      SUBROUTINE GAP (ICON,LINK,LANE,ITIME)
```

TABLE B-2, cont.

```
      COMMON /A1/ IREM,INEXT,IMAX,ISIZE,IFRSTCR(84,3),ILSTCAR(84,3),VEL(
     11500),POSN(1500),ACCEL(1500),ICAR(1500),NXTCAR(1500),IPRECAR(1500)
      COMMON /GEOM/ CURB(84),DBNDRY(84,6)
      COMMON /OPP/ IOPP(84,5)
      COMMON /SIG/ LCYCLE(84),NOPHAS(84,3),NPHASE(84,3,5),LPHASE(84,3,5)
     1,LOFFST(84,3)
      COMMON /RELAT/ LINCAR(84),NOLINK,LANCAR(84,3),NOLANE(84)
      COMMON /MSK/ MASK(20)
      IBITM(IX,M,N)=ISHFTR(IX,M).AND.MASK(N)
      DIMENSION TABLE1(15), TABLE2(15)
      INTEGER OFFPT
      DATA (TABLE1(I),I=1,15)/.10,.23,.42,.60,.76,.86,.92,.96,.98,.99,.9
     195,.9975,.999,.99945,.9997/
      DATA (TABLE2(I),I=1,15)/.01,.06,.20,.40,.62,.80,.90,.96,.982,.993,
     1.9975,.9991,.9997,.9998,.9999/
      RN=RANF(0)
      I=IOPP(LINK,3)
      IF (I.EQ.0) GO TO 170
      NOLLL=NOLANE(I)
      DO 170 LAN=1,NOLLL
      IF (LANCAR(I,LAN).EQ.0) GO TO 170
      NO=IFRSTCR(I,LAN)
  100 IF (NO.EQ.0) GO TO 170
      IF (POSN(NO).GT.CURB(I)+35.) GO TO 160
      IB=IBITM(ICAR(NO),33,3)
      IF (IB.GT.4) GO TO 160
      IF (POSN(NO).GT.CURB(I)+20.) GO TO 180
      X=CURB(I)-POSN(NO)+20.
      IF (ACCEL(NO).LE.0.) GO TO 130
  110 T3I=VEL(NO)*VEL(NO)+2.*ACCEL(NO)*X
      T=ABS(VEL(NO)-SQRT(T3I))/ACCEL(NO)
      GO TO 140
  120 XF=POSN(NO)-VEL(NO)*VEL(NO)/(2.*ACCEL(NO))
      IF (XF.LE.CURB(I)+20.) GO TO 170
      GO TO 110
  130 IF (VEL(NO).EQ.0.) GO TO 170
      IF (ACCEL(NO).LT.0.) GO TO 120
      T=X/VEL(NO)
  140 CONTINUE
      IF (T.LT.3.) GO TO 180
      IF (T.GT.10.) GO TO 170
      IT=2.*T-5.
      NCYC=(ITIME-LOFFST(LINK,LANE))/(LCYCLE(LINK))
      OFFPT=NCYC*LCYCLE(LINK)+LOFFST(LINK,LANE)
      IF (ITIME-OFFPT.LE.2) GO TO 150
      IF (TABLE1(IT).LE.RN) GO TO 180
      GO TO 170
  150 IF (TABLE2(IT).LE.RN) GO TO 180
      GO TO 170
  160 CONTINUE
      NO=NXTCAR(NO)
      GO TO 100
  170 CONTINUE
      ICON=1
C
C     ICON=1 MEANS ACCEPTABLE GAP
C
      RETURN
  180 CONTINUE
      ICON=2
C
```

TABLE B-2, cont.

```
C     ICON=2 MEANS UNACCEPTABLE GAP
C
      RETURN
C
      END


      SUBROUTINE BOOK (LINK,LANE,NUM,ITIME)
      COMMON /A1/ IREM,INEXT,IMAX,ISIZE,IFRSTCR(84,3),ILSTCAR(84,3),VEL(
     11500),POSN(1500),ACCEL(1500),ICAR(1500),NXTCAR(1500),IPRECAR(1500)
      COMMON /CAR/ IACC,TVEL,ITURN,EL,IACCL
      COMMON /PARAM/ A0,FREEK,REACT,CYCLE
      COMMON /GEOM/ CURB(84),DBNDRY(84,6)
      COMMON /BUK/ NHOLD(84,3)
      COMMON /B2/ TTIME(84),TDIST(84),TRTIME(15,105),TTTIME,TTDIST,HSTO(
     115,3,14)
      COMMON /B3/ DELAY(84),DELDEV(84),NCALL(84)
      COMMON /TRCE/ NTRACE,LKTRCE(15),NOTRC(15),INTTRC,KNTRAC(15)
      COMMON /KWNTZ/ NCAR(84),NOUTPT,ISTART,IEND,KGEN(84)
      COMMON /MSK/ MASK(20)
      DIMENSION HQ(105)
      IBITM(IX,M,N)=ISHFTR(IX,M).AND.MASK(N)
      REAL LNKTIM
      IF (POSN(NUM).LT.DBNDRY(LINK,ITURN)) GO TO 140
      NHOLD(LINK,LANE)=NHOLD(LINK,LANE)+1
      NCAR(LINK)=NCAR(LINK)+1
      LNKTIM=FLOAT(ITIME-IBITM(ICAR(NUM),0,13))
      TTTIME=TTTIME+LNKTIM
      START=FLOAT(IBITM(ICAR(NUM),14,9))
      IF (LINK.GE.ISTART) START=-START
      TTDIST=TTDIST+POSN(NUM)-START
      TTIME(LINK)=TTIME(LINK)+LNKTIM
      TDIST(LINK)=TDIST(LINK)+POSN(NUM)-START
      UNDEL=(POSN(NUM)-START)/TVEL
      DEL=LNKTIM-UNDEL
      CALL STIX5 (DELAY(LINK),DELDEV(LINK),DEL,NCALL(LINK),VMEAN,STDEV)
      IPOS=INT(POSN(NUM)-DBNDRY(LINK,ITURN))
      CALL JBITM (ICAR(NUM),14,9,IPOS)
      DO 100 I=1,NTRACE
      IF (LINK.EQ.LKTRCE(I)) GO TO 110
  100 CONTINUE
      GO TO 140
  110 IF (IBITM(ICAR(NUM),56,1).EQ.0) GO TO 140
      KNTRAC(I)=KNTRAC(I)+1
      DO 120 J=1,105
  120 HQ(J)=TRTIME(I,J)
      CALL STIX2 (HQ,200.,0.,100,LNKTIM,11HTRAVEL TIME,11,LINK)
      DO 130 J=1,105
  130 TRTIME(I,J)=HQ(J)
  140 CONTINUE
      RETURN
C
      END


      SUBROUTINE OUTPT (ITIME)
      COMMON /A1/ IREM,INEXT,IMAX,ISIZE,IFRSTCR(84,3),ILSTCAR(84,3),VEL(
     11500),POSN(1500),ACCEL(1500),ICAR(1500),NXTCAR(1500),IPRECAR(1500)
      COMMON /Z0/ IHOLE(1500),LASTONE
```

TABLE B-2, cont.

```
      COMMON /GEOM/ CURB(84),DBNDRY(84,6)
      COMMON /RELAT/ LINCAR(84),NOLINK,LANCAR(84,3),NOLANE(84)
      COMMON /TURN/ NREC(84,6),LREC(84,6),PTURN(84,6),PTRN(84,2)
      COMMON /BUK/ NHOLD(84,3)
      COMMON /QU/ LQUEUE(15,3),LQ(10),NQ
      COMMON /B2/ TTIME(84),TDIST(84),TRTIME(15,105),TTTIME,TTDIST,HSTO(
     115,3,14)
      COMMON /TRCE/ NTRACE,LKTRCE(15),NOTRC(15),INTTRC,KNTRAC(15)
      COMMON /KWNT7/ NCAR(84),NOUTPT,ISTART,IEND,KGEN(84)
      COMMON /CAPACTY/ LNCAP
      COMMON /PARAM2/ TVELMN(84,2),TVELSIG(84,2),ACCMN(2),ACCSIG(2),DECM
     1N(2),DECSIG(2),ELMN(2),ELSIG(2),PRTRUCK(84)
      COMMON /MSK/ MASK(20)
      DIMENSION HQ(45)
      REAL NORMAL
      IBITM(IX,M,N)=ISHFTR(IX,M).AND.MASK(N)
      DIMENSION PTR(6)
      DO 270 LINK=1,NOLINK
      NOLNN=NOLANE(LINK)
      DO 270 LANE=1,NOLNN
      IF (NHOLD(LINK,LANE).EQ.0) GO TO 270
      NHHH=NHOLD(LINK,LANE)
      DO 260 NUM=1,NHHH
      INDEX=IFRSTCR(LINK,LANE)
      IF=NXTCAR(INDEX)
      IF (LINK.LE.NOUTPT) GO TO 200
      ITURN=IBITM(ICAR(INDEX),33,3)
      POSN(INDEX)=POSN(INDEX)-DBNDRY(LINK,ITURN)
      N=NREC(LINK,ITURN)
      L=LREC(LINK,ITURN)
      LINCAR(N)=LINCAR(N)+1
      LANCAR(N,L)=LANCAR(N,L)+1
      LNCAR=LANCAR(N,L)
      CALL JBITM (ICAR(INDEX),0,13,ITIME)
      IL=ILSTCAR(N,L)
      IF (IL.EQ.0) GO TO 100
      NXTCAR(IL)=INDEX
  100 ILSTCAR(N,L)=INDEX
      IPRECAR(INDEX)=IL
      NXTCAR(INDEX)=0
      IF (LANCAR(N,L).EQ.1) IFRSTCR(N,L)=INDEX
      RN=RANF(0)
      IF (NOLANE(LINK).GT.1) GO TO 120
      DO 110 I=1,6
      JTURN=I
      IF (RN.LE.PTURN(LINK,I)) GO TO 140
  110 CONTINUE
      GO TO 140
  120 PTR(1)=PTURN(N,1)
      PTR(2)=PTURN(N,2)
      PTR(3)=PTRN(N,L)
      PTR(4)=PTURN(N,4)
      PTR(5)=PTURN(N,5)
      PTR(6)=PTURN(N,6)
      IF (ITURN.NE.3.AND.ITURN.NE.4) PTR(3)=PTURN(N,3)
      DO 130 I=1,6
      JTURN=I
      IF (RN.LE.PTR(I)) GO TO 140
  130 CONTINUE
  140 CONTINUE
      DO 150 I=1,NTRACE
```

TABLE B-2, cont.

```
      IF (N.EQ.LKTRCE(I).AND.LINK.EQ.LKTRCE(I-1)) GO TO 160
  150 CONTINUE
      CALL JBITM (ICAR(INDEX),56,1,0)
      GO TO 190
  160 IF (I+1.GT.NTRACE) GO TO 170
      IF (NREC(N,JTURN).EQ.LKTRCE(I+1)) GO TO 170
      CALL JBITM (ICAR(INDEX),56,1,0)
      GO TO 190
  170 IF (INTTRC.EQ.NOTRC(I)) GO TO 180
      NOTRC(I)=NOTRC(I)+1
      CALL JBITM (ICAR(INDEX),56,1,0)
      GO TO 190
  180 NOTRC(I)=0
      CALL JBITM (ICAR(INDEX),56,1,1)
  190 CALL JBITM (ICAR(INDEX),33,3,JTURN)
      IVEHTYP=IBITM(ICAR(INDEX),54,2)
      ITARVEL=NORMAL(TVELMN(N,IVEHTYP),TVELSIG(N,IVEHTYP))
      CALL JBITM (ICAR(INDEX),26,7,ITARVEL)
      CALL JBITM (ICAR(INDEX),36,4,7)
  200 CONTINUE
      LINCAR(LINK)=LINCAR(LINK)-1
      LANCAR(LINK,LANE)=LANCAR(LINK,LANE)-1
      IF (LANCAR(LINK,LANE).EQ.0) ILSTCAR(LINK,LANE)=0
      IF (LANCAR(LINK,LANE).EQ.0) IFRSTCR(LINK,LANE)=0
      DO 210 I=1,NQ
      IF (LINK.EQ.LQ(I)) GO TO 220
  210 CONTINUE
      GO TO 230
  220 IF (LQUEUE(I,LANE).LE.0) GO TO 230
      LQUEUE(I,LANE)=LQUEUE(I,LANE)-1
  230 CONTINUE
      IF (LANCAR(LINK,LANE).EQ.0) GO TO 240
      IFRSTCR(LINK,LANE)=IF
      IPRECAR(IF)=0
  240 IF (LANCAR(LINK,LANE).EQ.0) ILSTCAR(LINK,LANE)=0
      IF (LINK.GT.NOUTPT) GO TO 260
      LASTONE=LASTONE+1
      IF (LASTONE.GT.ISIZE) GO TO 250
      IHOLE(LASTONE)=INDEX
  250 CONTINUE
  260 CONTINUE
  270 CONTINUE
      DO 290 I=1,NQ
      L=LQ(I)
      N=NOLANE(L)
      DO 290 LN=1,N
      DO 280 J=1,14
  280 HQ(J)=HSTQ(I,LN,J)
      QLEN=FLOAT(LQUEUE(I,LN))
      CALL STIX2 (HQ,10.,0.,10,QLEN,12HQUEUE LENGTH,12,LINK)
      DO 290 J=1,14
  290 HSTQ(I,LN,J)=HQ(J)
      RETURN
C
      END
```

```
      SUBROUTINE QUEUE (LINK,LANE)
      COMMON /A1/ IREM,INEXT,IMAX,ISIZE,IFRSTCR(84,3),ILSTCAR(84,3),VEL(
     11500),POSN(1500),ACCEL(1500),ICAR(1500),NXTCAR(1500),IPRECAR(1500)
```

TABLE B-2, cont.

```
      COMMON /QU/ LQUEUE(15,3),LQ(10),NLQ
      COMMON /RELAT/ LINCAR(84),NOLINK,LANCAR(84,3),NOLANE(84)
      COMMON /MSK/ MASK(20)
      IBITM(IX,M,N)=ISHFTR(IX,M).AND.MASK(N)
      DO 100 I=1,NLQ
      IF (LINK.EQ.LQ(I)) GO TO 110
  100 CONTINUE
      GO TO 160
  110 CONTINUE
      IF (LQUEUE(I,LANE).EQ.0) GO TO 150
      LLQ=LQUEUE(I,LANE)
      IF=IFRSTCR(LINK,LANE)
      IF (LLQ.GT.1) GO TO 120
      NQ=NXTCAR(IF)
      GO TO 140
  120 DO 130 INTE=2,LLQ
  130 IF=NXTCAR(IF)
      IF (IF.EQ.ILSTCAR(LINK,LANE)) GO TO 160
      NQ=NXTCAR(IF)
  140 CONTINUE
      IF (NQ.EQ.0) GO TO 160
      IQ=IBITM(ICAR(NQ),36,4)
      IF (IQ.NE.1.AND.IQ.NE.3.AND.IQ.NE.6) GO TO 160
      LQUEUE(I,LANE)=LQUEUE(I,LANE)+1
      IF (NQ.EQ.ILSTCAR(LINK,LANE)) GO TO 160
      NQ=NXTCAR(NQ)
      GO TO 140
  150 CONTINUE
      IF (LANCAR(LINK,LANE).EQ.0) GO TO 160
      NQ=IFRSTCR(LINK,LANE)
      GO TO 140
  160 CONTINUE
      RETURN
C
      END


      SUBROUTINE SUMARY (VOL,ITIME,SUMKNT)
      COMMON /RELAT/ LINCAR(84),NOLINK,LANCAR(84,3),NOLANE(84)
      COMMON /B2/ TTIME(84),TDIST(84),TRTIME(15,105),TTTIME,TTDIST,HSTQ(
     115,3,14)
      COMMON /B3/ DELAY(84),DELDEV(84),NCALL(84)
      COMMON /TRCE/ NTRACE,LKTRCE(15),NOTRC(15),INTTRC,KNTRAC(15)
      COMMON /KWNTZ/ NCAR(84),NOUTPT,ISTART,IEND,KGEN(84)
      COMMON /QU/ LQUEUE(15,3),LQ(10),NQ
      COMMON /QPLOTQ/ DUM(5),IPLOT
      COMMON /TIMKK/ INITIM,LRUN,MINITIM
      INTEGER SUMKNT
      DIMENSION HQ(105)
      DIMENSION VOL(84)
      IPLOT=0
      KTIM=(ITIME-INITIM)/60
      WRITE (6,340) SUMKNT,KTIM
      WRITE (6,350)
      DO 140 LINK=1,NOLINK
      WRITE (6,360) LINK,NCAR(LINK)
      IF (LINK.LT.ISTART) GO TO 100
      WRITE (6,370) KGEN(LINK)
  100 DO 110 LK=1,NTRACE
      IF (LINK.EQ.LKTRCE(LK)) GO TO 120
```

TABLE B-2, cont.

```
110 CONTINUE
    GO TO 130
120 WRITE (6,380) KNTRAC(LK)
130 WRITE (6,390)
140 CONTINUE
    IF (TTTIME.EQ.0.) TTTIME=1.
    SPEED=TTDIST/TTTIME/1.47
    WRITE (6,400)
    NV=0
    DO 150 I=1,NOLINK
150 NV=NV+NCAR(I)
    TVM=TTDIST/5280.
    TD=0.
    DO 160 I=1,NOLINK
160 TD=DELAY(I)+TD
    ATD=TD/FLOAT(NV)
    ATDM=TD/TVM
    TTR=TTTIME/TVM
    SUMSQ=0.
    DO 170 I=1,NOLINK
170 SUMSQ=SUMSQ+DELDEV(I)
    STDEV=SQRT(ABS(SUMSQ/FLOAT(NV)-ATD*ATD))
    ATT=TTTIME/FLOAT(NV)
    WRITE (6,410) NV,TVM,TD,ATD,STDEV,ATDM,TTTIME,ATT,TTR,SPEED
    WRITE (6,420)
    DO 260 L=1,NOLINK
    CALL STIX6 (DELAY(L),DELDEV(L),DUM1,NCALL(L),VMEAN,STDEV)
    IF (TTIME(L).EQ.0.) GO TO 180
    SPEED=TDIST(L)/TTIME(L)/1.47
    GO TO 190
180 SPEED=0.
190 IF (NCAR(L).EQ.0) GO TO 200
    TIM=TTIME(L)/FLOAT(NCAR(L))
    GO TO 210
200 TIM=0.
210 TVM=TDIST(L)/5280.
    IF (TDIST(L).EQ.0.) GO TO 220
    ATDM=DELAY(L)/TVM
    TTR=TTIME(L)/TVM
    GO TO 230
220 ATDM=0.
    TTR=0.
230 IF (NCAR(L).EQ.0) GO TO 240
    ATT=TTIME(L)/FLOAT(NCAR(L))
    GO TO 250
240 ATT=0.
250 WRITE (6,410) NCAR(L),TVM,DELAY(L),VMEAN,STDEV,ATDM,TTIME(L),ATT,T
   1TR,SPEED
260 WRITE (6,430) L
    DO 290 LK=1,NTRACE
    LNKE=LKTRCE(LK)
    IF (KNTRAC(LK).EQ.0) GO TO 280
    DO 270 I=1,105
270 HQ(I)=TRTIME(LK,I)
    CALL STIX3 (HQ,200.,0.,100,LNKTIM,11HTRAVEL TIME,11,LNKE,0)
    GO TO 290
280 WRITE (6,440) LNKE
290 CONTINUE
    DO 330 L=1,NQ
    LK=LQ(L)
    N=NOLANE(LK)
```

TABLE B-2, cont.

```
      DO 330 LN=1,N
      DO 300 I=1,12
  300 T=T+HSTQ(L,LN,I)
      IF (T.EQ.0.) GO TO 320
      DO 310 J=1,14
  310 HQ(J)=HSTQ(L,LN,J)
      CALL STIX3 (HQ,10.,0.,10,DUM,12HQUEUE LENGTH,12,LK,LN)
      GO TO 330
  320 WRITE (6,450) LK
  330 CONTINUE
      RETURN
C
  340 FORMAT (1H1,4X,125(1H*),/5X,125(1H*),///60X, 11HSUMMARY NO.,I3,//5
     19X,  6HTIME=,I3,  9H MINUTES,///5X,125(1H*),/5X,125(1H*))
  350 FORMAT (////////23X, 14HNUM. GENERATED,14X, 11HEXIT VOLUME,10X, 12H
     1TRACE VOLUME,///)
  360 FORMAT (1H+,9X,  4HLINK,I3,36X,I5)
  370 FORMAT (1H+,26X,I5)
  380 FORMAT (1H+,74X,I5)
  390 FORMAT (1H )
  400 FORMAT (1H2,/////63X,  7HAVERAGE,4X,  5HTOTAL,6X.  7HAVERAGE,4X,
     16HTRAVEL,/13X,  5HTOTAL,6X,  5HTOTAL,5X,  5HTOTAL,6X,  7HAVERAGE,6
     2X,  5HDELAY,6X,  5HDELAY,5X,  6HTRAVEL,5X,  6HTRAVEL,6X,  4HTIME,5
     3X,  7HAVERAGE,/11X,  8HVEHICLES,4X,  7HVEHICLE,4X,  6HDELAY,7X,  5
     4HDELAY,5X,  9HSTD. DEV.,4X,  5HSEC./,6X,  4HTIME,7X,  4HTIME,6X,
     55HSEC./,6X,  5HSPEED,/11X,  9HPROCESSED,4X,  5HMILES,4X,  8HVEH.SE
     6C.,3X,  9HSEC./VEH.,3X,  9HSEC./VEH.,3X,  7HVEH.MI.,3X,  8HVEH.SEC
     7.,3X,  9HSEC./VEH.,3X,  7HVEH.MI.,5X,  3HMPH,//)
  410 FORMAT (13X,I5,4X,F8.3,2X,F7.0,5X,F8.3,5X,F7.3,3X,F8.3,3X,F7.0,4X,
     1F7.2,4X,F7.2,5X,F5.2)
  420 FORMAT (1H+,7HOVERALL,//)
  430 FORMAT (1H+,4HLINK,I3)
  440 FORMAT (1H4,5X,5(1H*), 19H   TOTAL CARS, LINK,I4,  5H ,= 0,5X,5(1H
     1*))
  450 FORMAT (1H4,5X,5(1H*), 22H   CARS IN QUEUE, LINK,I4,  5H ,= 0,5X,5
     1(1H*))
C
      END
```

```
      SUBROUTINE JBITM (X,M,N,Y)
      COMMON /MSK/ MASK(20)
      INTEGER X1,X,Y,Y1
      X1=ISHFTR(X,M).AND.MASK(N)
      X1=ISHFTL(X1,M)
      Y1=Y.AND.MASK(N)
      X=ISHFTL(Y1,M)+X-X1
      RETURN
C
      END
```

```
      SUBROUTINE STIX1 (TABLE,VMAX,VMIN,NINK,VAR,ITITLE,LNTHTL,LINK)
      COMMON /OPLOTO/ OPLOT,TDATE,ISH,IOVER,IUNDER,PLOTSW
      DIMENSION TABLE(110)
      INDEX=NINK+4
      TDATE=DATE(0)
      DO 100 K=1,INDEX
  100 TABLE(K)=0.0
```

TABLE B-2, cont.

```
      RETURN
C
      END


      SUBROUTINE STIX2 (TABLE,VMAX,VMIN,NINK,VAB,ITITLE,LNTHTL,LINK)
      DIMENSION ITITLE(LNTHTL), TABLE(NINK)
      TABLE(NINK+3)=TABLE(NINK+3)+VAB
      TABLE(NINK+4)=TABLE(NINK+4)+VAB*VAB
      VINK=(VMAX-VMIN)/NINK
      IF (VAB-VMIN) 100,110,110
  100 CONTINUE
      IV=0
      GO TO 120
  110 CONTINUE
      IV=(VAB-VMIN)/VINK
      IF (IV.GT.NINK) IV=NINK+1
  120 CONTINUE
      TABLE(IV+1)=TABLE(IV+1)+1
      RETURN
C
      END


      SUBROUTINE STIX3 (TABLE,VMAX,VMIN,NINK,VAB,ITITLE,LNTHTL,LINK,LANE
     1)
      COMMON /Y/ EPSILON,ALLOFIT
      COMMON /OPLOTO/ OPLOT,TDATE,ISW,IOVER,IUNDER,PLOTSW
      DIMENSION XTABLE(310), TABL(310)
      DIMENSION ITITLE(LNTHTL), TABLE(NINK)
      INTEGER OUTPUT,PUNCH,CORESZ,WORDSZ,PLOTSW,WDSZ
      REAL IOVER,IUNDER
      EQUIVALENCE (WDSZ,WORDSZ)
      DATA INPUT,OUTPUT,PUNCH/5,6,7/
      DATA WDSZ/10/
      DATA EPSILON,ALLOFIT/.0005,99.9995/
      DO 100 I=1,310
      XTABLE(I)=0.0
      TABL(I)=XTABLE(I)
  100 CONTINUE
      SUMS=0.0
      T=0.0
      VINK=(VMAX-VMIN)/NINK
      INDEX=NINK+2
      DO 110 I=1,INDEX
  110 T=T+TABLE(I)
      IF (T) 120,130,120
  120 CONTINUE
      IUNDER=100.0*TABLE(1)/T
      IOVER=100.0*TABLE(INDEX)/T
      CMEAN=TABLE(NINK+3)/T
      STD=SQRT(ABS(TABLE(NINK+4)/T-CMEAN*CMEAN))
      GO TO 140
  130 CONTINUE
      IUNDER=0
      IOVER=0
      CMEAN=0.0
      STD=0.0
  140 CONTINUE
```

TABLE B-2, cont.

```
      TT=TABLE(1)+TABLE(NINK+1)
      A2=CFLOT+1.0
      WRITE (OUTPUT,340)
      WRITE (OUTPUT,320) LINK
      IF (LANE.EQ.0) GO TO 150
      WRITE (OUTPUT,330) LANE
  150 CONTINUE
      IF (FLOTSW.NE.0) WRITE (OUTPUT,310) A2
      INDEXT=LNTHTL/WDSZ+1
      IF (MOD(LNTHTL,WDSZ).EQ.0) INDEXT=INDEXT-1
      IF (INDEXT.EQ.0) GO TO 160
      WRITE (OUTPUT,260) (ITITLE(I),I=1,INDEXT)
  160 CONTINUE
      WRITE (OUTPUT,240) T,CMEAN,STD
      WRITE (OUTPUT,300)
      II=-1
      DO 200 I=1,INDEX
      II=II+3
      XTABLE(II)=(TABLE(I)/T)*100.0
      TABL(II+1)=(I-1.5)*VINK+VMIN
      TABL(II)=TABL(II+1)
      TABL(II-1)=TABL(II)
      SUMS=SUMS+XTABLE(II)
      IF (SUMS.LT.EPSILON)IGC+TOI200
      UL=(I-1)*VINK+VMIN
      CUMR=100.0-SUMS
      CMEAN1=UL/CMEAN
      IF (STD) 170,180,170
  170 CONTINUE
      DEVMEAN=(UL-CMEAN)/STD
      GO TO 190
  180 CONTINUE
      DEVMEAN=INFIN
  190 CONTINUE
      PB=QPLQPR(CMEAN,UL,STD)
      WRITE (OUTPUT,250) UL,TABLE(I),XTABLE(II),SUMS,CUMR,CMEAN1,DEVMEAN
     1,PB
      IF (I.EQ.1) WRITE (OUTPUT,290)
      IF (I.EQ.INDEX) WRITE (OUTPUT,280)
      IF (SUMS.GE.ALLOFIT.AND.I.LT.INDEX-1) GO TO 210
  200 CONTINUE
      GO TO 230
  210 CONTINUE
      WRITE (OUTPUT,270)
      J=I+1
      DO 220 I=J,INDEX
      II=II+3
      TABL(II+1)=(I-1.5)*VINK+VMIN
      TABL(II)=TABL(II+1)
      TABL(II-1)=TABL(II)
  220 CONTINUE
  230 CONTINUE
      RETURN
C
  240 FORMAT (//,20X, 16HENTRIES IN TABLE,5X, 13HMEAN ARGUMENT,5X,  8HST
     1ANDARD, 10H DEVIATION,/,26X,F10.0,8X,F10.3,13X,F10.3,//)
  250 FORMAT (10X,F14.3,F15.0,6F15.3)
  260 FORMAT (//,20X,12A10)
  270 FORMAT (10X, 34HREMAINING FREQUENCIES ARE ALL ZERO)
  280 FORMAT ( 24H+OVERFLOW  XXXXXXXXXXXXX)
  290 FORMAT ( 10H+UNDERFLOW)
```

TABLE B-2, cont.

```
  300 FORMAT (  1H0,18X,  5HVALUE,7X,  8HOBSERVED,7X,  9HPER CENT,2(5X,
     110HCUMULATIVE),7X,  8HMULTIPLE,6X,  9HDEVIATION,5X, 10HCUMULATIVE,
     2/,30X,  9HFREQUENCY,7X,  8HOF TOTAL,5X, 10HPERCENTAGE,6X,  9HREMAI
     3NDER,8X,  7HOF MEAN,6X,  9HFROM MEAN,4X, 11HPROBABILITY,/)
  310 FORMAT (40X, 40HTHIS DATA IS ACCOMPANIED BY GRAPH NUMBER,F10.0)
  320 FORMAT (////20X,  4HLINK,I3)
  330 FORMAT (///20X,  4HLANE,I3)
  340 FORMAT (1H1)
C
      END


      FUNCTION ORLOPB(Z,AVG,DEV)
      COMP=-1.0
      IF (DEV) 110,100,110
  100 CONTINUE
      U=0
      GO TO 120
  110 CONTINUE
      U=ABS(Z-AVG)/DEV
  120 CONTINUE
      IF (U-1.6) 130,130,160
  130 TERM=U*0.3989423
      GINT=TERM
      U=-U*U
      R=0.0
  140 R=R+2.
      TERM=TERM*U*(R-1.)/(R*(R+1.))
      GINT=GINT+TERM
      IF (ABS(TERM)-.0001) 150,150,140
  150 GINT=.5+GINT*COMP*SIGN(1.,(Z-AVG))
      GO TO 190
  160 PHI=U
      UCO=1.
      QQ=1.
      R=0.
      Q=0.
      ALT=-1.
  170 R=R+1.
      UCO=1./(R*UCO)
      TERM=Q
      Q=QQ
      QQ=Q*U*UCO+TERM
      ALT=-ALT
      TERM=ALT/(Q*QQ)
      PHI=PHI+TERM
      IF (ABS(TERM)-.0005) 180,180,170
  180 GINT=.3989423*EXP(-.5*U*U)/PHI
      GINT=.5+(.5-GINT)*COMP*SIGN(1.,(Z-AVG))
  190 CONTINUE
      ORLOPB=GINT
      RETURN
C
      END


      SUBROUTINE STIX4 (TABLE1,TABLE2,VAR,NCALL,VMEAN,STDEV)
      TABLE1=0.
      TABLE2=0.
```

TABLE B-2, cont.

```
      NCALL=0
      RETURN
C
      END


      SUBROUTINE STIX5 (TABLE1,TABLE2,VAR,NCALL,VMEAN,STDEV)
      TABLE1=TABLE1+VAR
      TABLE2=TABLE2+VAR*VAR
      NCALL=NCALL+1
      RETURN
C
      END


      SUBROUTINE STIX6 (TABLE1,TABLE2,VAR,NCALL,VMEAN,STDEV)
      T=FLOAT(NCALL)
      IF (T.LE.0.) GO TO 100
      VMEAN=TABLE1/T
      STDEV=SQRT(ABS(TABLE2/T-VMEAN*VMEAN))
      RETURN
  100 VMEAN=0.
      STDEV=0.
      RETURN
C
      END


      REAL FUNCTION NORMAL(A,B)
      DIMENSION TABLE(80)
      DIMENSION NAME(2)
      DATA TABLE/.0000,.01994,.03983,.05962,.07926,.09871,.11791,.13683,
     1.15542,.17364,.19146,.20884,.22575,.24215,.25804,.27337,.28814,.30
     2234,.31579,.32894,.34134,.35314,.36433,.37493,.38493,.39435,.40320
     3,.41149,.41924,.42647,.43319,.43943,.44520,.45053,.45543,.45994,.4
     46407,.46784,.47128,.47441,.47725,.47982,.48214,.48422,.48610,.4877
     58,.48928,.49061,.49180,.49286,.49379,.49461,.49534,.49598,.49653,.
     649702,.49744,.49781,.49813,.49841,.49865,.49886,.49903,.49918,.499
     731,.49942,.49952,.49960,.49966,.49972,.49977,.49981,.49984,.49987,
     8.49989,.49991,.49993,.49994,.49995,.49996/
      SIGN=+1.
      Z=RANF(0)
      IF (Z.LT.0.5) GO TO 100
      Z=Z-.5
      SIGN=-1.
  100 DO 110 I=1,80
      IF (Z.LT.TABLE(I)) GO TO 120
  110 CONTINUE
  120 CONTINUE
      NORMAL=A+SIGN*B*(FLOAT(I)-RANF(0)-1.0)*0.05
      RETURN
C
      END


      REAL FUNCTION NEGEXP(A)
      NEGEXP=0.75-(1.0/A-0.75)*ALOG(RANF(0))
```

TABLE B-2, cont.

```
      RETURN                                                            AE   30
C                                                                       AE   40
      END                                                               AE   50



      FUNCTION TIMELP(NPRINT)                                           AF   10
      DATA ISW,X,Y/0,0.0,0.0/                                           AF   20
      IF (ISW.NE.0) GO TO 100                                           AF   30
      ISW=-1                                                            AF   40
      CALL SECOND (Y)                                                   AF   50
  100 CONTINUE                                                          AF   60
      CALL SECOND (X)                                                   AF   70
      Z=X-Y                                                             AF   80
      IF (NPRINT.EQ.0) GO TO 110                                        AF   90
      WRITE (6,120) Z                                                   AF  100
  110 CONTINUE                                                          AF  110
      Y=X                                                               AF  120
      TIMELP=Z                                                          AF  130
      RETURN                                                            AF  140
C                                                                       AF  150
  120 FORMAT (/, 35H      ..CP TIME SINCE LAST CALL ...,F8.3,  8H SECOND AF  160
     1S,/)                                                              AF  170
C                                                                       AF  180
      END                                                               AF  190
```

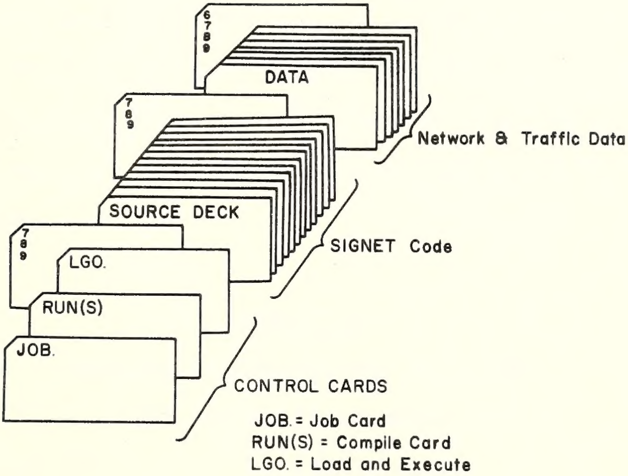


FIGURE B-2

SIGNET Program Deck Structure

Appendix C

Traffic Flow Data

The following tables give detailed summaries of traffic characteristics during average peak hour conditions on the test network. The data is presented for each of the six 15-minute periods comprising the study period, 4:00 to 5:30 P.M. Traffic volumes, turning probabilities, and lane distributions were obtained through a system of machine and manual counts. Traffic signal settings were obtained by stop-watch.

TABLE C-1

Network Traffic Signal Settings

(All times in Seconds)

Link Number	Indication	Duration	Cycle Length	Offset[1]
26	R	32	70	49
	G	35		
	A	3		
27	R	23	60	45
	G + GAL	7		
	G	27		
	A	3		

---

[1] All offsets measured from beginning of red, Link 54 (State Street and Grant Street), to beginning of first phase listed.

Link Number	Indication	Duration	Cycle Length	Offset[1]
28	R	19	60	26
	G + GAL	9		
	G	29		
	A	3		
29	G	33	60	53
	A	3		
	R	24		
30	R	36	65	31
	G + GAL	5		
	G	21		
	A	3		
31	G	20	65	0
	A	3		
	R	42		
32	G	55	55	0
33	R	20	55	39
	G	32		
	A	3		
34	G	55	55	0
35	G	29	60	17
	A	4		
	R	27		
36	G	28	75	60
	A	3		
	R	44		
37	G	60	60	0
38	G	33	60	53
	A	3		
	R	24		
39	R	31	60	26
	G	26		
	A	3		
40	G	20	60	45
	A	3		
	R	37		

Link Number	Indication	Duration	Cycle Length	Offset
41	R (All red)	2	70	49
	G	15		
	A	3		
	R	50		
42	R	25	50	24
	G	22		
	A	3		
43	R	36	60	53
	G	21		
	A	3		
44	G	55	55	0
45	G	55	55	0
46	R	20	55	39
	G	32		
	A	3		
47	G	20	65	0
	A	3		
	R	42		
48	R	33	60	17
	G	24		
	A	3		
49	R	33	60	17
	G	24		
	A	3		
50	R	20	65	64
	G	42		
	A	3		
51	R	32	65	40
	G	31		
	A	3		
52	R	23	65	53
	G	38		
	A	4		

Link Number	Indication	Duration	Cycle Length	Offset
53	R	35	65	26
	G	27		
	A	3		
54	R	36	65	0
	G	26		
	A	3		
55	R	31	75	60
	GAR	30		
	G	11		
	A	3		
56	R	23	65	0
	G + GAL	10		
	G	3		
	A	29		
57	R	20	65	26
	G + GAL	12		
	G	30		
	A	3		
58	GAL	20	65	53
	A	3		
	GAS	38		
	A	4		
59	R	32	65	40
	G	30		
	A	3		
60	G	17	65	64
	A	3		
	R	45		
61	R	20	65	64
	G	42		
	A	3		
62	G	17	65	64
	A	3		
	R	45		

Link Number	Indication	Duration	Cycle Length	Offset
63	G	29	65	40
	A	3		
	R	33		
64	G	29	65	40
	A	3		
	R	33		
65	G	17	65	26
	A	3		
	R	45		
66	G	17	65	26
	A	3		
	R	45		
67	G + GAL	8	65	31
	G	35		
	A	3		
	R	29		
68	R	41	65	31
	G	21		
	A	3		
69	R	8	65	31
	G	25		
	A	3		
	R	29		
70	GADR	31	75	60
	GAL + GADR	27		
	A	3		
	R	14		
71	R	61	75	60
	G	11		
	A	3		
72	G	17	55	39
	A	3		
	R	35		
73	G	17	55	39
	A	3		
	R	35		

Link Number	Indication	Duration	Cycle Length	Offset
74	R	55	55	0
75	G	30	60	17
	A	3		
	R	27		
76	R	33	60	17
	G	24		
	A	3		
77	R	36	60	53
	G	21		
	A	3		
78	G	16	60	26
	A	3		
	R	41		
79	G	20	60	45
	A	3		
	R	37		
80	R	33	60	45
	G	24		
	A	3		
81	R	20	70	49
	G	9		
	A	3		
	R	38		
82	R	32	70	49
	G	35		
	A	3		
83	G	22	50	24
	A	3		
	R	25		
84	G	22	50	24
	A	3		
	R	25		

TABLE C-2

TABULATION OF TRAFFIC CHARACTERISTICS, AVERAGE CONDITIONS

4.00 - 4.15 PM

LINK NUMBER	TRAFFIC FLOW RATE VPH	TURNING PROBABILITIES 1	2	3	4	5	6	LANE DISTRIBUTION 1	2	3
NETWORK LINKS										
26	---	.12	--	.45	.39	--	.04	.57	.43	--
27	---	.05	--	.34	.33	--	.28	.39	.33	.28
28	---	.05	--	.37	.35	--	.23	.42	.35	.23
29	---	.10	--	.46	.32	--	.12	.56	.44	--
30	---	.28	--	.44	--	--	.28	.72	.28	--
31	---	.50	--	.27	.23	--	--	.77	.23	--
32	---	--	--	.53	.18	--	.29	.53	.47	--
33	---	.05	--	.50	.45	--	--	.55	.45	--
34	---	--	--	.53	.47	--	--	.53	.47	--
35	---	--	--	.53	.46	--	.01	.53	.47	--
36	---	.24	--	--	--	.76	--	.24	.76	--
37	---	--	--	.42	--	.35	.23	.42	.35	.23
38	---	.17	--	.42	.41	--	--	.59	.41	--
39	---	.06	--	.49	.37	--	.08	.55	.45	--
40	---	.43	--	.26	--	--	.31	.43	.26	.31
41	---	--	.18	.20	--	--	.62	.18	.20	.62
42	---	.15	--	.43	.24	--	.18	.58	.42	--
43	---	.27	--	.39	--	--	.34	.66	.34	--
44	---	--	--	.53	.47	--	--	.53	.47	--
45	---	--	--	.53	.35	--	.12	.53	.47	--
46	---	.27	--	.27	.46	--	--	.54	.46	--
47	---	.28	--	.35	.37	--	--	.63	.37	--
48	---	--	--	.29	--	--	.71	.29	.71	--
49	---	.23	--	.38	.39	--	--	.61	.39	--
50	---	.18	--	.39	.34	--	.09	.57	.43	--

TABLE C-2. CONT.

LINK NUMBER	TRAFFIC FLOW RATE VPH	TURNING PROBABILITIES 1	2	3	4	5	6	LANE DISTRIBUTION 1	2	3
51	---	--	--	.53	.41	--	.06	.53	.47	--
52	---	.34	--	.66	--	--	--	.34	.66	--
53	---	.01	--	.52	.47	--	--	.53	.47	--
54	---	.10	--	.33	.57	--	--	.43	.57	--
55	---	.79	--	--	.13	--	.08	.79	.21	--
56	---	.03	--	.39	.37	--	.21	.42	.37	.21
57	---	--	--	.50	.46	--	.04	.50	.46	.04
58	---	--	--	.48	.39	--	.13	.48	.39	.13
59	---	.02	--	.51	.47	--	--	.53	.47	--
INPUT LINKS										
60	216	.26	--	.11	--	--	.63	.37	.63	--
61	472	.01	--	.52	.42	--	.05	.53	.47	--
62	128	.38	--	.15	--	--	.47	.53	.47	--
63	356	.17	--	.13	--	--	.70	.17	.13	.70
64	116	.64	--	--	--	--	.36	.64	.36	--
65	52	.54	--	--	--	--	.46	.54	.46	--
66	256	.45	--	.10	--	--	.45	.45	.10	.45
67	340	.01	--	.45	--	--	.54	.46	.54	--
68	84	.04	--	.80	--	--	.16	.84	.16	--
69	140	.24	--	.69	--	--	.07	.93	.07	--
70	680	--	.25	--	--	--	.75	.25	--	.75
71	136	.22	--	.75	--	--	.03	1.00	--	--
72	180	.45	--	--	--	--	.55	.45	.55	--
73	108	.54	--	--	--	--	.46	.54	.46	--
74	0	1.00	--	--	--	--	--	1.00	--	--
75	484	.08	--	.47	.45	--	--	.55	.45	--
76	36	1.00	--	--	--	--	--	1.00	--	--

TABLE C-7. CONT.

LINK NUMBER	TRAFFIC FLOW RATE VPH	TURNING PROBABILITIES						LANE DISTRIBUTION		
		1	2	3	4	5	6	1	2	3
77	208	.03	--	.53	--	--	.44	.56	.44	--
78	168	.53	--	.19	--	--	.28	.72	--	.28
79	218	.09	--	.77	--	--	.14	.86	.14	--
80	366	.18	--	.43	.34	--	.05	.61	.34	.05
81	12	.67	.33	--	--	--	--	1.00	--	--
82	216	.06	--	.48	.42	--	.04	.54	.46	--
83	56	.12	--	.88	--	--	--	1.00	--	--
84	108	--	--	.33	--	--	.67	.33	.67	--

TABLE C-3
TABULATION OF TRAFFIC CHARACTERISTICS, AVERAGE CONDITIONS
4.15 - 4.30 PM

LINK NUMBER	TRAFFIC FLOW RATE VPH	TURNING PROBABILITIES						LANE DISTRIBUTION		
		1	2	3	4	5	6	1	2	3
NETWORK LINKS										
26	---	.16	--	.41	.39	--	.04	.57	.43	--
27	---	.09	--	.37	.39	--	.23	.42	.35	.23
28	---	.04	--	.49	.35	--	.12	.53	.35	.12
29	---	.02	--	.51	.34	--	.13	.53	.47	--
30	---	.41	--	.37	--	--	.22	.78	.22	--
31	---	.51	--	.26	.23	--	--	.77	.23	--
32	---	--	--	.53	.18	--	.29	.53	.47	--
33	---	.05	--	.50	.45	--	--	.55	.45	--
34	---	--	--	.53	.47	--	--	.53	.47	--
35	---	--	--	.53	.46	--	.01	.53	.47	--
36	---	.24	--	--	--	.76	--	.24	.76	--
37	---	--	--	.39	--	.34	.27	.39	.34	.27
38	---	.21	--	.40	.38	--	.01	.61	.39	--
39	---	.03	--	.51	.42	--	.04	.54	.46	--
40	---	.46	--	.28	--	--	.26	.46	.28	.26
41	---	.18	.20	--	--	--	.62	.38	--	.62
42	---	.06	--	.49	.30	--	.15	.55	.45	--
43	---	.16	--	.40	--	--	.44	.56	.44	--
44	---	--	--	.53	.47	--	--	.53	.47	--
45	---	--	--	.53	.35	--	.12	.53	.47	--
46	---	.27	--	.27	.46	--	--	.54	.46	--
47	---	.32	--	.22	.46	--	--	.54	.46	--
48	---	--	--	.19	--	--	.81	.19	.81	--
49	---	.29	--	.35	.36	--	--	.64	.36	--
50	---	.18	--	.39	.34	--	.09	.57	.43	--

TABLE C-3. CONT.

LINK NUMBER	TRAFFIC FLOW RATE VPH	TURNING PROBABILITIES 1	2	3	4	5	6	LANE DISTRIBUTION 1	2	3
51	---	--	--	.53	.41	--	.06	.53	.47	--
52	---	.33	--	.67	--	--	--	.33	.67	--
53	---	.06	--	.49	.45	--	--	.55	.45	--
54	---	.12	--	.31	.57	--	--	.43	.57	--
55	---	.80	--	--	.13	--	.07	.80	.20	--
56	---	.03	--	.41	.42	--	.14	.44	.42	.14
57	---	--	--	.50	.45	--	.05	.50	.45	.05
58	---	--	--	.46	.38	--	.16	.46	.38	.16
59	---	.07	--	.46	.47	--	--	.53	.47	--
INPUT LINKS										
60	204	.26	--	.11	--	--	.63	.37	.63	--
61	464	.07	--	.48	.39	--	.06	.55	.45	--
62	108	.38	--	.15	--	--	.47	.53	.47	--
63	492	.17	--	.11	--	--	.72	.17	.11	.72
64	132	.68	--	--	--	--	.32	.68	.32	--
65	116	.54	--	--	--	--	.46	.54	.46	--
66	388	.45	--	.10	--	--	.45	.45	.10	.45
67	232	--	--	.61	--	--	.39	.61	.39	--
68	88	.03	--	.85	--	--	.12	.88	.12	--
69	88	.34	--	.60	--	--	.06	.94	.06	--
70	604	--	.25	--	--	--	.75	.25	--	.75
71	88	.22	--	.75	--	--	.03	1.00	--	--
72	188	.45	--	--	--	--	.55	.45	.55	--
73	112	.54	--	--	--	--	.46	.54	.46	--
74	0	1.00	--	--	--	--	--	1.00	--	--
75	544	.07	--	.48	.45	--	--	.55	.45	--
76	40	1.00	--	--	--	--	--	1.00	--	--

TABLE C-3. CONT.

LINK NUMBER	TRAFFIC FLOW RATE VPH	TURNING PROBABILITIES 1	2	3	4	5	6	LANE DISTRIBUTION 1	2	3
77	276	.04	--	.49	--	--	.47	.53	.47	--
78	188	.67	--	.11	--	--	.22	.78	--	.22
79	216	.33	--	.51	--	--	.16	.84	.16	--
80	320	.21	--	.37	.38	--	.04	.58	.38	.04
81	64	.38	-.07	--	--	--	.69	.31	.69	--
82	460	.04	--	.50	.37	--	.09	.54	.46	--
83	120	.12	--	.88	--	--	--	1.00	--	--
84	140	--	--	.31	--	--	.69	.31	.69	--

TABLE C-4
TABULATION OF TRAFFIC CHARACTERISTICS, AVERAGE CONDITIONS
4.30 - 4.45 PM

LINK NUMBER	TRAFFIC FLOW RATE VPH	TURNING PROBABILITIES						LANE DISTRIBUTION		
		1	2	3	4	5	6	1	2	3
NETWORK LINKS										
26	---	.17	--	.42	.39	--	.02	.59	.41	--
27	---	.02	--	.51	.17	--	.30	.53	.17	.30
28	---	.04	--	.48	.36	--	.12	.52	.36	.12
29	---	.08	--	.48	.38	--	.06	.56	.44	--
30	---	.35	--	.30	--	--	.35	.65	.35	--
31	---	.54	--	.23	.23	--	--	.77	.23	--
32	---	--	--	.53	.18	--	.29	.53	.47	--
33	---	.05	--	.56	.39	--	--	.61	.39	--
34	---	--	--	.53	.47	--	--	.53	.47	--
35	---	--	--	.53	.46	--	.01	.53	.47	--
36	---	.11	--	--	--	.89	--	.11	.89	--
37	---	--	--	.35	--	.38	.27	.35	.38	.27
38	---	.16	--	.43	.41	--	--	.59	.41	--
39	---	.03	--	.52	.41	--	.04	.55	.45	--
40	---	.49	--	.25	--	--	.26	.49	.25	.26
41	---	--	.52	.30	--	--	.18	.52	.30	.18
42	---	.17	--	.48	.27	--	.18	.57	.43	--
43	---	.10	--	.44	--	--	.46	.54	.46	--
44	---	--	--	.53	.47	--	--	.53	.47	--
45	---	--	--	.53	.24	--	.23	.53	.47	--
46	---	.12	--	.39	.49	--	--	.51	.49	--
47	---	.30	--	.34	.36	--	--	.64	.36	--
48	---	--	--	.15	--	--	.85	.15	.85	--
49	---	.35	--	.31	.34	--	--	.66	.34	--
50	---	.15	--	.42	.37	--	.06	.57	.43	--

TABLE C-4. CONT.

LINK NUMBER	TRAFFIC FLOW RATE VPH	TURNING PROBABILITIES 1	2	3	4	5	6	LANE DISTRIBUTION 1	2	3
51	---	--	--	.50	.46	--	.04	.50	.50	--
52	---	.29	--	.71	--	--	--	.29	.71	--
53	---	.04	--	.50	.46	--	--	.54	.46	--
54	---	.06	--	.49	.45	--	--	.55	.45	--
55	---	.81	--	.10	--	--	.09	.91	.09	--
56	---	.03	--	.43	.40	--	.14	.46	.40	.14
57	---	--	--	.51	.45	--	.04	.51	.45	.04
58	---	--	--	.56	.33	--	.11	.56	.33	.11
59	---	.07	--	.46	.47	--	--	.53	.47	--
INPUT LINKS										
60	144	.22	--	.09	--	--	.69	.31	.69	--
61	304	.01	--	.52	.42	--	.05	.53	.47	--
62	100	.31	--	.42	--	--	.27	.73	.27	--
63	364	.18	--	.19	--	--	.63	.18	.19	.63
64	172	.60	--	--	--	--	.40	.60	.40	--
65	108	.52	--	--	--	--	.48	.52	.48	--
66	288	.32	--	.04	--	--	.64	.32	.04	.64
67	272	--	--	.48	--	--	.52	.48	.52	--
68	128	.07	--	.76	.07	--	.10	.83	.17	--
69	172	.40	--	.56	--	--	.04	.96	.04	--
70	680	.01	.33	--	--	--	.66	.34	--	.66
71	140	.16	--	.84	--	--	--	1.00	--	--
72	240	.60	--	--	--	--	.40	.60	.40	--
73	180	.44	--	--	--	--	.56	.44	.56	--
74	0	--	--	1.00	--	--	--	1.00	--	--
75	612	.07	--	.48	.45	--	--	.55	.45	--
76	36	1.00	--	--	--	--	--	1.00	--	--

TABLE C-4. CONT.

LINK NUMBER	TRAFFIC FLOW RATE VPH	TURNING PROBABILITIES						LANE DISTRIBUTION		
		1	2	3	4	5	6	1	2	3
77	280	.04	--	.42	--	--	.54	.46	.54	--
78	308	.67	--	.11	--	--	.22	.78	--	.22
79	236	.26	--	.63	--	--	.11	.89	.11	--
80	484	.11	--	.45	.41	--	.03	.56	.41	.03
81	48	.62	.15	--	--	--	.23	.77	.23	--
82	396	.04	--	.50	.42	--	.04	.54	.46	--
83	88	.30	--	.70	--	--	--	1.00	--	--
84	104	--	--	.17	--	--	.83	.17	.83	--

TABLE C-5

TABULATION OF TRAFFIC CHARACTERISTICS, AVERAGE CONDITIONS

4.45 - 5.00 PM

LINK NUMBER	TRAFFIC FLOW RATE VPH	TURNING PROBABILITIES						LANE DISTRIBUTION		
		1	2	3	4	5	6	1	2	3
NETWORK LINKS										
26	---	.11	--	.46	.42	--	.01	.57	.43	--
27	---	.06	--	.49	.26	--	.19	.55	.26	.19
28	---	.01	--	.49	.33	--	.17	.50	.33	.17
29	---	.06	--	.49	.31	--	.14	.55	.45	--
30	---	.27	--	.48	--	--	.25	.75	.25	--
31	---	.39	--	.38	.23	--	--	.77	.23	--
32	---	--	--	.53	.20	--	.27	.53	.47	--
33	---	.10	--	.51	.39	--	--	.61	.39	--
34	---	--	--	.53	.47	--	--	.53	.47	--
35	---	--	--	.53	.45	--	.02	.53	.47	--
36	---	.25	--	--	--	.75	--	.25	.75	--
37	---	--	--	.33	--	.39	.28	.33	.39	.28
38	---	.17	--	.42	6.41	--	--	.59	.41	--
39	---	.06	--	.51	.43	--	--	.57	.43	--
40	---	.46	--	.26	--	--	.28	.46	.26	.28
41	---	--	.53	.31	--	--	.16	.53	.31	.16
42	---	.08	--	.48	.35	--	.09	.56	.44	--
43	---	.29	--	.40	--	--	.31	.69	.31	--
44	---	--	--	.53	.47	--	--	.53	.47	--
45	---	--	--	.53	.29	--	.18	.53	.47	--
46	---	.06	--	.45	.49	--	--	.51	.49	--
47	---	.36	--	.31	.33	--	--	.67	.33	--
48	---	--	--	.20	--	--	.80	.20	.80	--
49	---	.36	--	.30	.34	--	--	.66	.34	--
50	---	.15	--	.42	.34	--	.09	.57	.43	--

TABLE C-5, CONT.

LINK NUMBER	TRAFFIC FLOW RATE VPH	TURNING PROBABILITIES 1	2	3	4	5	6	LANE DISTRIBUTION 1	2	3
51	---	--	--	.52	.46	--	.02	.52	.48	--
52	---	.40	--	.60	--	--	--	.40	.60	--
53	---	.08	--	.47	.45	--	--	.55	.45	--
54	---	.04	--	.50	.46	--	--	.54	.46	--
55	---	.82	--	.10	--	--	.08	.92	.08	--
56	---	.03	--	.43	.39	--	.15	.46	.39	.15
57	---	--	--	.49	.44	--	.07	.49	.44	.07
58	---	--	--	.56	.37	--	.07	.56	.37	.07
59	---	.01	--	.52	.47	--	--	.53	.47	--
INPUT LINKS										
60	200	.19	--	.31	--	--	.50	.50	.50	--
61	536	.01	--	.52	.44	--	.03	.53	.47	--
62	192	.50	--	.20	--	--	.30	.70	.30	--
63	348	.08	--	.22	--	--	.70	.08	.22	.70
64	312	.63	--	--	--	--	.37	.63	.37	--
65	124	.47	--	--	--	--	.53	.47	.53	--
66	372	.45	--	.07	--	--	.48	.45	.07	.48
67	340	.06	--	.52	--	--	.42	.58	.42	--
68	172	.04	--	.85	--	--	.11	.89	.11	--
69	148	.41	--	.42	--	--	.17	.83	.17	--
70	748	.01	.28	--	--	--	.71	.29	--	.71
71	112	.22	--	.78	--	--	--	1.00	--	--
72	276	.56	--	--	--	--	.44	.56	.44	--
73	328	.40	--	--	--	--	.60	.40	.60	--
74	0	--	--	1.00	--	--	--	1.00	--	--
75	612	.04	--	.50	.46	--	--	.54	.46	--
76	52	1.00	--	--	--	--	--	1.00	--	--

TABLE C-5. CONT.

LINK NUMBER	TRAFFIC FLOW RATE VPH	TURNING PROBABILITIES						LANE DISTRIBUTION		
		1	2	3	4	5	6	1	2	3
77	340	.02	--	.56	--	--	.42	.58	.42	--
78	580	.65	--	.12	--	--	.23	.77	--	.23
79	164	.15	--	.80	--	--	.05	.95	.05	--
80	460	.20	--	.40	.36	--	.04	.60	.36	.04
81	32	.29	.14	--	--	--	.57	.43	.57	--
82	392	.01	--	.57	.43	--	--	.53	.47	--
83	148	.20	--	.80	--	--	--	1.00	--	--
84	188	--	--	.37	--	--	.63	.37	.63	--

TABLE C-6

TABULATION OF TRAFFIC CHARACTERISTICS, AVERAGE CONDITIONS

5.00 - 5.15 PM

LINK NUMBER	TRAFFIC FLOW RATE VPH	TURNING PROBABILITIES						LANE DISTRIBUTION		
		1	2	3	4	5	6	1	2	3
NETWORK LINKS										
26	---	.17	--	.42	.41	--	--	.59	.41	--
27	---	.04	--	.50	.27	--	.19	.54	.27	.19
28	---	--	--	.50	.34	--	.16	.50	.34	.16
29	---	.06	--	.49	.38	--	.07	.55	.45	--
30	---	.33	--	.38	--	--	.29	.71	.29	--
31	---	.47	--	.30	.23	--	--	.77	.23	--
32	---	--	--	.53	.06	--	.41	.53	.47	--
33	---	.02	--	.59	.39	--	--	.61	.39	--
34	---	--	--	.53	.47	--	--	.53	.47	--
35	---	--	--	.53	.46	--	.01	.53	.47	--
36	---	.14	--	--	--	.86	--	.14	.86	--
37	---	--	--	.34	--	.38	.28	.34	.38	.28
38	---	.18	--	.41	.41	--	--	.59	.41	--
39	---	.04	--	.53	.42	--	.01	.57	.43	--
40	---	.35	--	.29	--	--	.36	.35	.29	.36
41	---	--	.48	.37	--	--	.15	.48	.37	.15
42	---	.12	--	.45	.37	--	.06	.57	.43	--
43	---	.15	--	.51	--	--	.34	.66	.34	--
44	---	--	--	.53	.47	--	--	.53	.47	--
45	---	--	--	.53	.32	--	.15	.53	.47	--
46	---	.07	--	.44	.49	--	--	.51	.49	--
47	---	.27	--	.34	.39	--	--	.61	.39	--
48	---	--	--	.24	--	--	.76	.24	.76	--
49	---	.43	--	.26	.31	--	--	.69	.31	--
50	---	.07	--	.50	.34	--	.09	.57	.43	--

TABLE C-6. CONT.

LINK NUMBER	TRAFFIC FLOW RATE VPH	TURNING PROBABILITIES 1	2	3	4	5	6	LANE DISTRIBUTION 1	2	3
51	---	--	--	.50	.45	--	.05	.50	.50	--
52	---	.34	--	.66	--	--	--	.34	.66	--
53	---	.04	--	.50	.46	--	--	.54	.46	--
54	---	.12	--	.45	.43	--	--	.57	.43	--
55	---	.76	--	.15	--	--	.09	.91	.09	--
56	---	.03	--	.44	.41	--	.12	.47	.41	.12
57	---	--	--	.48	.44	--	.08	.48	.44	.08
58	---	--	--	.56	.27	--	.17	.56	.27	.17
59	---	.06	--	.50	.44	--	--	.56	.44	--
INPUT LINKS										
60	232	.16	--	.29	--	--	.55	.45	.55	--
61	408	.03	--	.51	.42	--	.04	.54	.46	--
62	252	.23	--	.22	--	--	.55	.45	.55	--
63	568	.21	--	.24	--	--	.55	.21	.24	.55
64	396	.64	--	--	--	--	.36	.64	.36	--
65	240	.55	--	--	--	--	.45	.55	.45	--
66	672	.44	--	.11	--	--	.45	.44	.11	.45
67	384	.02	--	.37	--	--	.61	.39	.61	--
68	228	.12	--	.82	--	--	.06	.94	.06	--
69	136	.20	--	.57	--	--	.23	.77	.23	--
70	628	--	.30	--	--	--	.70	.30	--	.70
71	100	.23	--	.77	--	--	--	1.00	--	--
72	212	.61	--	--	--	--	.39	.61	.39	--
73	332	.42	--	--	--	--	.58	.42	.58	--
74	0	--	--	1.00	--	--	--	1.00	--	--
75	748	.04	--	.50	.46	--	--	.54	.46	--
76	24	1.00	--	--	--	--	--	1.00	--	--

TABLE C-6. CONT.

LINK NUMBER	TRAFFIC FLOW RATE VPH	TURNING PROBABILITIES						LANE DISTRIBUTION		
		1	2	3	4	5	6	1	2	3
77	356	.03	--	.45	--	--	.52	.48	.52	--
78	628	.54	--	.14	--	--	.32	.68	--	.32
79	272	.30	--	.58	--	--	.12	.88	.12	--
80	412	.22	--	.40	.35	--	.03	.62	.35	.03
81	52	.47	.26	--	--	--	.27	.73	.27	--
82	380	.05	-.05	.51	.43	--	.06	.51	.49	--
83	204	.04	--	.96	--	--	--	1.00	--	--
84	116	--	--	.42	--	--	.58	.42	.58	--

TABLE C-7

TABULATION OF TRAFFIC CHARACTERISTICS, AVERAGE CONDITIONS

5.15 - 5.30 PM

LINK NUMBER	TRAFFIC FLOW RATE VPH	TURNING PROBABILITIES 1	2	3	4	5	6	LANE DISTRIBUTION 1	2	3
NETWORK LINKS										
26	---	.16	--	.43	.41	--	--	.59	.41	--
27	---	.04	--	.50	.27	--	.19	.54	.27	.19
28	---	.02	--	.48	.34	--	.16	.50	.34	.16
29	---	.09	--	.47	.32	--	.12	.56	.44	--
30	---	.34	--	.44	--	--	.22	.78	.22	--
31	---	.56	--	.21	.23	--	--	.77	.23	--
32	---	--	--	.53	.13	--	.34	.53	.47	--
33	---	.01	--	.60	.39	--	--	.61	.39	--
34	---	--	--	.53	.47	--	--	.53	.47	--
35	---	--	--	.53	.45	--	.02	.53	.47	--
36	---	.24	--	--	--	.76	--	.24	.76	--
37	---	--	--	.29	--	.41	.30	.29	.41	.30
38	---	.12	--	.45	.42	--	.01	.57	.43	--
39	---	.09	--	.47	.41	--	.03	.56	.44	--
40	---	.35	--	.29	--	--	.36	.35	.29	.36
41	---	--	.47	.32	--	--	.21	.47	.32	.21
42	---	.12	--	.45	.31	--	.12	.57	.43	--
43	---	.12	--	.55	--	--	.33	.67	.33	--
44	---	--	--	.53	.47	--	--	.53	.47	--
45	---	--	--	.53	.34	--	.13	.53	.47	--
46	---	.07	--	.44	.49	--	--	.51	.49	--
47	---	.30	--	.34	.36	--	--	.64	.36	--
48	---	--	--	.29	--	--	.71	.29	.71	--
49	---	.19	--	.41	.40	--	--	.60	.40	--
50	---	.13	--	.44	.39	--	.04	.57	.43	--

TABLE C-7. CONT.

LINK NUMBER	TRAFFIC FLOW RATE VPH	TURNING PROBABILITIES						LANE DISTRIBUTION		
		1	2	3	4	5	6	1	2	3
51	---	--	--	.50	.45	--	.05	.50	.50	--
52	---	.36	--	.64	--	--	--	.36	.64	--
53	---	.05	--	.49	.46	--	--	.54	.46	--
54	---	.06	--	.49	.45	--	--	.55	.45	--
55	---	.78	--	.11	--	--	.11	.89	.11	--
56	---	.05	--	.39	.38	--	.18	.44	.38	.18
57	---	--	--	.49	.44	--	.07	.49	.44	.07
58	---	--	--	.56	.32	--	.12	.56	.32	.12
59	---	.03	--	.50	.47	--	--	.53	.47	--
INPUT LINKS										
60	152	.31	--	.19	--	--	.50	.50	.50	--
61	444	.04	--	.50	.42	--	.04	.54	.46	--
62	156	.22	--	.29	--	--	.49	.51	.49	--
63	424	.17	--	.16	--	--	.67	.17	.16	.67
64	232	.52	--	--	--	--	.48	.52	.48	--
65	180	.77	--	--	--	--	.23	.77	.23	--
66	408	.28	--	.06	--	--	.66	.28	.06	.66
67	340	.04	--	.37	--	--	.59	.41	.59	--
68	160	.12	--	.84	--	--	.04	.96	.04	--
69	220	.43	--	.50	--	--	.07	.93	.07	--
70	796	--	.35	--	--	--	.65	.35	--	.65
71	108	.20	--	.80	--	--	--	1.00	--	--
72	192	.53	--	--	--	--	.47	.53	.47	--
73	228	.41	--	--	--	--	.59	.41	.59	--
74	0	--	--	1.00	--	--	--	1.00	--	--
75	472	.04	--	.50	.46	--	--	.54	.46	--
76	24	1.00	--	--	--	--	--	1.00	--	--

TABLE C-7. CONT.

LINK NUMBER	TRAFFIC FLOW RATE VPH	TURNING PROBABILITIES 1	2	3	4	5	6	LANE DISTRIBUTION 1	2	3
77	336	.06	--	.57	--	--	.37	.63	.37	--
78	400	.57	--	.12	--	--	.31	.69	--	.31
79	252	.30	--	.58	--	--	.12	.88	.12	--
80	388	.22	--	.40	.35	--	.03	.62	.35	.03
81	64	.33	.40	--	--	--	.27	.73	.27	--
82	320	.03	--	.51	.42	--	.04	.54	.46	--
83	156	.34	--	.66	--	--	--	1.00	--	--
84	168	--	--	.32	--	--	.68	.32	.68	--

Waddleworth